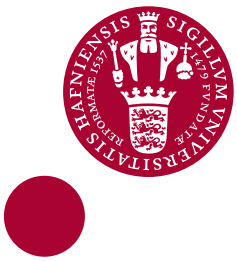


Proceedings - Second International Symposium on Foundations and Applications of Blockchain (FAB)

University of Southern California, Los Angeles, California, April 5, 2019

Barahmand, Sumita ; Ghandeharizadeh, Shahram ; Krishnamachari, Bhaskar ; Lugones, Diego; Nambiar, Raghunath ; Slaats, Tijs

Publication date:
2019

Document version
Publisher's PDF, also known as Version of record

Second International Symposium on Foundations and Applications of Blockchain

Proceedings

Sumita Barahmand
Shahram Ghandeharizadeh
Bhaskar Krishnamachari
Diego Lugones
Raghunath Nambiar
Tijs Slaats

University of Southern California
Los Angeles, California
April 5, 2019

Message from the Chairs

The second International Symposium on Foundations and Applications of Blockchain (FAB) brings together researchers and practitioners of blockchain to share and exchange research results. This one-day event is held at the beautiful campus of the University of Southern California, Los Angeles, CA, on April 5, 2019.

The program consists of four exciting refereed technical papers from around the world, a timely panel on the future of blockchain, and seven invited keynotes from academia and industry. The peer reviewed papers were provided with 3 to 4 reviews. This year, the evaluation process included a one week rebuttal period for the authors to respond to reviewer comments.

One of the peer-reviewed papers titled "BDML: Blockchain-based Distributed Machine Learning for Model Training and Evolution" stood out and was recognized with the best paper award. We congratulate the authors for their timely technical contribution.

We thank our international program committee, Gowri Sankar Ramachandran as the webmaster, and Brienne Jessica Moore for her logistical support of the event. Finally, we wish to thank the authors and keynote speakers for their contributions and the panelists for an exciting discussion of the future of blockchain.

Sumita Barahmand, Proceedings Chair
Shahram Ghandeharizadeh, Program Co-Chair
Bhaskar Krishnamachari, General Chair
Diego Lugones, Industrial Co-Chair
Raghunath Nambiar, Industrial Co-Chair
Tijs Slaats, Program Co-Chair

Table of Contents

FAB 2019 Conference Organization

General Chair: Bhaskar Krishnamachari (*University of Southern California*)

Program Chair: Shahram Ghandeharizadeh (*University of Southern California),* Tijs Slaats *(University of Copenhagen)*

Industrial chair: Diego Lugones *(Nokia-Bell Labs)*
Raghunath Nambiar *(AMD)*

Proceedings chair: Sumita Barahmand (*Microsoft*)

Web chair: Gowri Sankar Ramachandran (*University of Southern California*)

Program Committee Members:
Daniel Augot *(Inria, France)*
Mashael AlSabah *(Qatar Computing Research Institute)*
Sumita Barahmand *(Microsoft)*
Eric Chung *(DApperNetwork)*
Claudio di Ciccio *(Vienna University of Economics and Business)*
Søren Debois *(IT University of Copenhagen)*
Ajay Dholakia *(Lenovo Data Center Group)*
Eric Diehl *(Sony Pictures Entertainment)*
Marlon Dumas *(University of Tartu)*
Boris Düdder *(University of Copenhagen)*
Luciano GarcÍa-Bañuelos *(University of Tartu)*
Vincent Gramoli *(University of Sydney)*
Abdelkader Hameurlain *(Paul Sabatier University, Toulouse, France)*
Ming-Deh Huang *(USC)*
Zhiyuan Jiang *(Tsinghua University)*
Salil Kanhere *(UNSW)*
Yaron Kanza *(AT&T)*
Bhaskar Krishnamachari *(USC)*
Genevieve Leveille *(Otentic8)*
Chen Li *(UC Irvine)*
Ron van der Meyden *(University of New South Wales)*
Beng Chin Ooi *(National University of Singapore)*
Charalampos (Babis) Papamanthou *(University of Maryland - College Park)*
Fabio Porto *(Laboratório Nacional de Computação Científica – LNCC)*

Jason Potts *(RMIT University)*
Björn Scheuermann *(Humboldt University of Berlin)*
Peter Sestoft *(IT University of Copenhagen)*
Mark Staples *(CSIRO)*
Vassilis Tsotras *(UC Riverside)*
Sherry Xu *(CSIRO)*
Kiran Yedavalli *(Cisco)*

FAB 2019 Conference Sponsors

Gold Sponsors

SOVEREIGNWALLET

University Sponsors

USC Viterbi School of Engineering
Center for Cyber-Physical Systems and the Internet of Things
CCI

Ming Hsieh Institute
Ming Hsieh Department of Electrical Engineering

Publicity Partners

ELEV8

Demystifying Blockchains: Decentralized and Fault-Tolerant Storage for the Future of Big Data?

Amr El Abbadi
Department of Computer Science, University of California, Santa Barbara.
(in collaboration with: Divy Agrawal, Mohammad Amiri, Sujaya Maiyya, Victor Zakhary)

ABSTRACT

Bitcoin is a successful and interesting example of a global scale peer-to-peer cryptocurrency that integrates many techniques and protocols from cryptography, distributed systems, and databases. The main underlying data structure is blockchain, a scalable fully replicated structure that is shared among all participants and guarantees a consistent view of all user transactions by all participants in the cryptocurrency system. The novel aspect of Blockchain is that historical data about all transactions is maintained in the absence of any central authority. This property of Blockchain has given rise to the possibility that future applications will transition from centralized databases to a fully decentralized storage based on blockchains. In this talk, we start by developing an understanding of the basic protocols used in blockchain, and elaborate on their main advantages and limitations. To overcome these limitations, we will explore some of the challenges of managing large scale fully replicated ledgers in the context of achieving large scale consensus. Finally, we ponder over recent efforts to use blockchains in diverse applications.

BIO

Amr El Abbadi is a Professor of Computer Science at the University of California, Santa Barbara. He received his B. Eng. from Alexandria University, Egypt, and his Ph.D. from Cornell University. Prof. El Abbadi is an ACM Fellow, AAAS Fellow, and IEEE Fellow. He was Chair of the Computer Science Department at UCSB from 2007 to 2011. He has served as a journal editor for several database journals, including, The VLDB Journal, IEEE Transactions on Computers and The Computer Journal. He has been Program Chair for multiple database and distributed systems conferences. He currently serves on the executive committee of the IEEE Technical Committee on Data Engineering (TCDE) and was a board member of the VLDB Endowment from 2002 to 2008. In 2007, Prof. El Abbadi received the UCSB Senate Outstanding Mentorship Award for his excellence in mentoring graduate students. In 2013, his student, Sudipto Das received the SIGMOD Jim Gray Doctoral Dissertation Award. Prof. El Abbadi is also a co-recipient of the Test of Time Award at EDBT/ICDT 2015. He has published over 300 articles in databases and distributed systems and has supervised over 35 PhD students.

Blocks of Trust – Using the Blockchain to Build Trust in IoT

Erez Waisbard
Researcher at Bell Labs

ABSTRACT

The Internet of Things (IoT) carries a big promise, to improve our lives through numerous connected devices. At the same time, it also presents new security challenges. One of the biggest challenges for the Internet of Things (IoT) is the establishing of trust as the traditional PKI model does not fit with the heterogeneous IoT ecosystem. Another big challenge arises from the questionable security of many of these devices that are becoming targets of choice for numerous exploits. Exploits that leads to compromising the networks to which they connect. In this talk we will show how the blockchain can be used to build a new distributed trust model that answers these challenges.

BIO

Dr. Erez Waisbard is a researcher at Bell Labs. His main research interests include: Blockchain technology, IoT, Privacy, Networking and Algorithms. Before joining Bell Labs Erez was a security architect at Cisco\NDS where he specialized in IoT security and content protection. Erez holds a PhD from Bar-Ilan university and M.Sc from the Weizmann Institute of Science.

Enterprise Blockchain Is Transforming the Global Supply Chain Today!!

Gil Perez
Senior Vice President, Head of Digital Customer Initiatives at SAP

ABSTRACT

In this talk, Mr. Perez will describe the Enterprise Blockchain journey over 100 companies embark on with SAP across Agriculture, Seafood, CPG, Pharma and Retail in the last 36 months. From initial innovations concepts, thorough pilots to live production systems compliant to FDA regulations. What are the challenges that each industry/company faced? what were the drivers for the transformation and lesson learned? Where are we today? and how far are we from Blockchain becoming an integral part of any modern supply chain solution?

BIO

Gil Perez is SAP Senior Vice President, Head of Digital Customer Initiatives. Gil is a member of the SAP SE executive team and has lead SAP global Blockchain efforts across all of SAP Products & Innovations teams. Gil has over 20 years of experience in enterprise software and new product introduction for multiple vertical industries while working at SAP and numerous startups. Gil brings a rich and diverse background as a co-founder, investor and senior executive in 5 companies which were acquired. Three of the five companies were acquired by SAP AG. Gil is based in Palo Alto, CA

Economics and Blockchain: Beyond Tokens

Stephanie Hurder
Partner and Founding Economist of Prysm Group

ABSTRACT

In this talk, Dr. Stephanie Hurder will discuss why a holistic approach to economic design -- including insights from fields as diverse as social choice theory, game theory, contract theory, and market design -- is essential for blockchain organizations. Dr. Hurder will discuss how major tech companies such as Microsoft, Google, and eBay have leveraged these insights for over a decade to improve the performance of their marketplaces, and discuss areas of blockchain and economics that are ripe for future innovation.

BIO

Dr. Stephanie Hurder is a Partner and Founding Economist of Prysm Group, a firm that specializes in economic and governance design for distributed ledger-based projects. She is a frequent keynote speaker and lecturer on economics and DLT, including Consensus, SXSW, Polychain Capital, and the IBM Blockchain Accelerator. Her research on the economics of blockchain has been presented at Harvard, Stanford, UC Berkeley, DARPA, and the Federal Reserve of Cleveland. Dr. Hurder is an advisor to blockchain projects at the World Economic Forum, a Visiting Scholar at the Center for Cyber-Physical Systems and the Internet of Things at USC, and a contributor to the MIT Cryptoeconomics Lab blog. Prior to co-founding Prysm Group, Dr. Hurder held economics research positions at MIT Sloan and Merrill Lynch. While a management consultant at the Boston Consulting Group, Dr. Hurder was selected as an Ambassador to the BCG Henderson Institute and co-authored multiple publications on organizational effectiveness and design. Dr. Hurder holds an AB in Mathematics Phi Beta Kappa and magna cum laude, an AM in Economics, and a PhD in Business Economics from Harvard University.

On the Need for Formal and Effective Internal Governance Mechanisms in Permissionless Blockchains

David Galindo
Head of Cryptography, Fetch.AI

ABSTRACT

With the birth and rise of cryptocurrencies such as Bitcoin, and its advanced variant Ethereum offering added functionalities such as a smart-contracts, the concept of blockchain has been popularised. Blockchains are software artefacts that bring the promise of facilitating radically new governmental and commercial applications, mainly thanks to their nature as trust-enabling tools, to the extent that commentators have claimed that improved societal organisation forms will be possible thanks to them, where human corruption and instability will be reduced to a minimum. Taking into consideration this ar-reaching goal, we perform a critical examination of the following question: can existing blockchain communities mechanisms and procedures for collective decision-making (which we refer to as internal blockchain governance), live up to those ambitions? By building on scholar literature on governance and law, we argue that if permissionless blockchain systems do not build internal governance mechanisms founded on formalised and effective social mechanisms, then they are unlikely to be taken up at scale as a tool for social coordination, and are thus likely to remain, at best, a marginal technology. This is based on joint work with Prof Karen Yeung, University of Birmingham.

BIO

Dr David Galindo is currently Head of Cryptography at Fetch.AI, a digital economics and artificial intelligence startup based in Cambridge UK, and Associate Professor in Computer Security at the University of Birmingham. David has more than 15 years of experience in applied cryptography research, both in academia and industry. His work has been published in top academic venues in computer security and has been deployed by governments around the globe.

Software Architecture and Engineering for Blockchain Applications

Ingo Weber
Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Sydney

ABSTRACT

Blockchain is a novel distributed ledger technology, which has attracted a wide range of interests for building the next generation of applications in almost all industry sectors. The broad range of applications is made possible by smart contracts, which transform a blockchain system into a general compute platform. For this new paradigm and technology platform, we investigated its impact on software architecture and engineering practices. Our starting point for this investigation was the question "what do architects and engineers need to know about blockchain to make good use of it?" In this keynote, I will cover the main insights from this work, recently summarized in the book "Architecture for Blockchain Applications", Springer, 2019.

BIO

Dr Ingo Weber is a Principal Research Scientist & Team Leader of the Architecture & Analytics Platforms (AAP) team at Data61, CSIRO in Sydney. In addition he is a Conjoint Associate Professor at UNSW Australia and an Adjunct Associate Professor at Swinburne University. He has published around 100 refereed papers and two books. A third book, "Architecture for Blockchain Applications", will be published by Springer in late 2018. The AAP team led by Dr Weber tackles major challenges around applications based on Blockchain and Distributed Ledger Technologies, approaching the topic from the areas of software architecture and engineering, business process management, and dependability.

Bringing Enterprise to Blockchain—Moving from Science Experiment to Business Mainstream

Sarabjeet (Jay) Singh
Senior Director of Products at Oracle

ABSTRACT

You hear about blockchain, IoT, and artificial intelligence at every turn, but how much is hype and how much is real? These technologies offer the potential to fundamentally alter the arc of business, jobs, and society at large. Blockchain offers the promise of increased trust and accountability; IoT enables unprecedented connectivity between devices, machines, and people; and artificial intelligence delivers prediction with precision. In this session explore the impact of these exciting technologies for businesses and learn about their value to enterprise applications. You will also hear about customers that are using blockchain, IoT, and AI to transform their businesses today.

BIO

Sarabjeet (Jay) Singh is a Senior Director of Products at Oracle, responsible for PaaS offerings in the area of Autonomous Cloud, Artificial Intelligence, Blockchain, and IoT. He has over 19 years of industry experience in Software Development, Product Management, and Marketing working in companies such as VMware, Pivotal Software, Cisco Systems, SunGard Availability Services, and many venture-backed Startups in SF Bay Area. He holds an MBA from Haas School of Business, U.C. Berkeley, M.S. Electrical Engineering from Virginia Tech, and B.Tech from NSIT (University of Delhi, India).

Future of Blockchain

Raghunath Nambiar *(AMD),* Gaurav Chawla *(Dell)*, Zak Cole *(Whiteblock)*,
Ajay Dholakia *(Lenovo Data Center Group)*, Eric Diehl *(Sony)*, David Galindo *(Fetch.AI)*

ABSTRACT

Blockchain is a top emerging area with opportunities in research and enterprise as well as industry verticals. This panel brings industry experts together to discuss various challenges associated with extending applications with blockchain. Our panellists will engage in exciting discussions about the future of blockchain in terms of the technology and the underlying systems but also with relation to use cases and industry impact. The panel will also discuss some of the emerging research in the area, and lessons learned in practice.

BIO

Raghunath Nambiar (AMD): Raghunath Nambiar is the Corporate Vice President and Chief Technology Officer of Datacenter Ecosystems and Application Engineering at AMD. He brings years of technical accomplishments with significant expertise in systems architecture, performance engineering, and creating disruptive technology solutions. Raghu has served in leadership positions on industry standards committees for performance evaluation, including elected chairman of the committees for artificial intelligence, big data systems, Internet of Things, and founding chair of TPC's International Conference Series on Performance Evaluation and Benchmarking. He has published more than 75 peer-reviewed papers and book chapters, 12 books in Lecture Series in Computer Science (LNCS), and holds ten patents with several pending. Prior to AMD, Raghu was the Chief Technology Officer of Cisco UCS business where he played a lead role in accelerating the growth of the Cisco UCS as top server platform. Also responsible for incubating the analytics portfolio, building a team, developing partnerships and go to market strategy, and growing it to a main stream business. Before joining Cisco, Raghu was an architect at Hewlett-Packard responsible for several industry-first and disruptive technology solutions and a decade of performance benchmark leadership. He holds master's degrees from University of Massachusetts and Goa University and completed an advanced management program from Stanford University. Raghu's recent book titled Transforming Industry Through Data Analytics examines the role of analytics in enabling digital transformation, how the explosion in internet connections affects key industries, and how applied analytics will impact our future.

Gaurav Chawla (Dell): Gaurav Chawla is a Fellow and Vice President in the Dell EMC Server and Infrastructure Systems Group, Office of the CTO. Gaurav has 25+ years of experience in various areas in computer industry and has been with Dell since 2005. He currently focuses on technology strategy and investigations for Dell EMC Server Group in the areas of Edge Computing (IoT and 5G) and Blockchain. For Blockchain, his primary focus is in the areas of permissioned blockchains and its applicability to various industry use cases. Prior to this he focused on Enterprise Storage/SDS, Hyper-converged Infrastructure, Private/Hybrid Cloud, HPC Storage, Telco NFV and Unix Kernel Projects. He has also contributed to multiple industry consortiums, standards and open source projects including IoT OpenFog Consortium, Linux Foundation OpenDaylight SDN Project, T11 FC/FCoE, IEEE DCB (Data Center Bridging) and SNIA CDMI standard. He has B.S and M.S. in Computer Engineering, and holds 50+ granted and pending patents.

Zak Cole (Whiteblock): Zak has been involved in the blockchain space since 2012 and Ethereum development since 2016. He received his training as a communications engineer in the United States Marine Corps where he and his unit were responsible for building and maintaining the communications and Internet infrastructure for Iraq's Al Anbar Province during Operation Iraqi Freedom 07-08. These efforts included the engineering of field data centers, satellite communications, and the implementation and management of mission-critical cryptographic assets. After retiring from the Marine Corps, Zak worked in the field of cyber security, network engineering, and applied cryptography. He also

ran an Internet agency and worked with Google prior to founding Whiteblock, a blockchain testing company whose clientele includes the Department of Defense, ConsenSys, Beam, RChain, Syscoin, and the Ethereum Foundation.

Ajay Dholakia (Lenovo): Ajay is a Principal Engineer, Senior Solution Architect and Chief Technologist for Software, Solutions and Networking Development within Lenovo Data Center Group. In this role, he is leading the development of customer solutions in the areas of AI, big data, analytics and cloud computing. He is also driving new projects for solution development using emerging technologies including Internet of Things (IoT) and blockchain. In his career spanning over 25 years, he has led diverse projects in research, technology, product and solution development and business/technical strategy. Ajay holds more than 50 patents and has authored over 40 technical publications including the book "Introduction to Convolutional Codes with Applications." Ajay earned a B. E. (Hons.) in Electrical and Electronics Engineering from the Birla Institute of Technology and Science in India, an MBA from the Henley Business School in the U.K. and M.S. and Ph.D. in Electrical and Computer Engineering from North Carolina State University, Raleigh, NC, USA. He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE) and a member of the Association for Computing Machinery (ACM).

Eric Diehl (Sony)

David Galindo (Fetch.AI): Dr David Galindo is currently Head of Cryptography at Fetch.AI, a digital economics and artificial intelligence startup based in Cambridge UK, and Associate Professor in Computer Security at the University of Birmingham. David has more than 15 years of experience in applied cryptography research, both in academia and industry. His work has been published in top academic venues in computer security and has been deployed by governments around the globe.

BDML: Blockchain-based Distributed Machine Learning for model training and evolution

Qigang Wang, Mei Li, Wanlu Zhang, Peng Wang, Zhongchao Shi, Feiyu Xu
AI Lab, Lenovo
{wangqg1, limei8, zhangwl12, wangpeng31, shizc2, fxu}@lenovo.com
[Type: research; Length: long]

ABSTRACT

In this paper, we present BDML, a blockchain-based architecture for collaborative development of AI models among different teams, in particular, among different organizations. Without the need to expose their private datasets, BDML enables researchers from different teams/organizations to jointly develop one model for a specific problem. BDML guarantees that the latest model in the blockchain is always the optimal model for that problem. BDML defines its own transaction types and a protocol for participants to collaborate, reach consensus, and earn the reward. We demonstrate that BDML can obtain model convergence and achieve better accuracy than any single participant can achieve alone. And BDML achieves the effect of the so-called super dataset (putting all participants' datasets together) training, which is usually unpractical. At the research community level, BDML eliminates redundant efforts by isolated AI researchers and helps to accumulate their efforts together.

Keywords

blockchain, deep learning, distributed training

1. INTRODUCTION

AI researchers develop AI models to solve problems. To develop a working model for a specific problem, they design the model structure, prepare a large amount of data, and spend a lot of computation power to train the model. The training process usually takes a long time. And this process is to some extent empirical. Many iterations may be needed to finally create a working model. There are many cases that researchers from different teams/organizations work on the same problem independently, with their own model structure and dataset. This has led to redundant efforts to build the same wheel. One of the key reasons for this situation is the data privacy issue. Taking the e-healthcare problem as an example, patient data in hospitals are very sensitive and not supposed to be shared. Hence each hospital has to develop their own model based on their own data. For rare diseases, it is challenging for one hospital to accumulate enough data and innovation becomes difficult.

There are already cooperation among researchers at different levels in different areas, e.g., sharing datasets [1–3], sharing model structures [4, 5], sharing pretrained models [6], sharing deep learning frameworks [7,8]. These activities provide an easier starting point and help to eliminate the redundant effort to some extent. However, existing methods either require data sharing or a centralized server coordinating the training process, which raise data privacy concern.

We propose a blockchain-based distributed machine learning architecture (BDML) for collaborative model training and evolution. The distributed consensus model of blockchain has motivated us to create BDML. This model generates a trusted database in an untrusted community. Although participants may or may not trust each other, they all trust the transactions on the blockchain. BDML targets at consortium blockchains, which grant control to a group of approved individuals or organizations. This architecture allows researchers in a consortium to share implicit knowledge about their data without exposing the data. Researchers from different organizations can jointly contribute to the same blockchain and get rewarded for their contribution. The result is that there is only one model being developed for a specific problem. And participants can benefit from this model according to their contributions. With BDML, there can be one consortium for one problem. The consortium consists of researchers from different organizations who have a common objective. They work together to keep optimizing the best model for that problem with a blockchain. There are many problems,

Second International Symposium on Foundations and Applications of Blockchain (FAB '19) April 5, 2019, Los Angeles, California, USA.

and so there can be many consortiums. BDML enables every consortium to work on a common problem without data privacy concern. By eliminating redundant effort, BDML saves resources at the society level.

We design and implement a simple scenario for BDML, where all participants keep using the same model structure and only update weight. We evaluate BDML's feasibility and applicability on image classification tasks. We do extensive experiments on the MNIST [1] digits recognition task with the LeNet-5 model [9]. To demonstrate BDML's applicability on more complex tasks, we also evaluate the CIFAR-10 [2] image classification task with the VGG-16 model [5] and the LiTS [3] liver segmentation task with the Unet model [10]. The experimental results show that BDML can obtain model convergence and achieve better accuracy than any single participant can achieve alone. And the accuracy of the model produced by BDML is close to the centralized case where the model is trained on the union of all participants' datasets.

The main contributions of this work are:

1. We propose a blockchain-based architecture for collaborative model training, which solves the data privacy concern and ensures trust.

2. We investigate its feasibility with a reference design and demonstrate its effectiveness.

2. BACKGROUND

BDML employs two existing technologies: blockchain and deep learning data parallelism distributed training. This section provides background information of these two technologies and presents related previous work.

2.1 Blockchain

Figure 1 shows a general structure of blockchain. It is a secure distributed immutable database of blocks linked in chronological order by hash pointers. Those blocks contain all transactions that have been executed and shared among a community. Each transaction in the blockchain is verified by all or majority of the community with a consensus rule. Once entered, information can not be changed. This technology was invented with Bitcoin by an anonymous person or group known as Satoshi Nakamoto [11]. Its application has gone far beyond Bitcoin today [12].

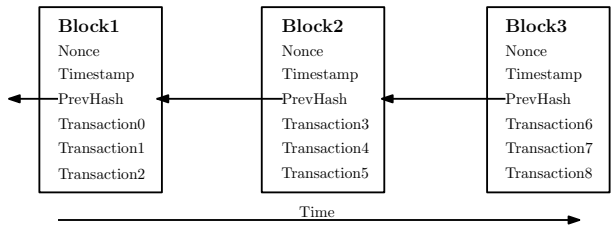


Figure 1: Blockchain

There are different varieties of blockchain : public, consortium and (fully) private. The typical features of blockchains are as follows:

- *Participant*: members of a blockchain network.

- *Miner*: participants who compete to create new blocks in order to earn the reward.

- *Asset*: anything that can be owned or controlled to produce value.

- *Transaction*: a (conditional) transfer of asset that is broadcast to the network and collected into blocks.

- *Block*: a blockchain entry consisting of a group of transactions.

- *Proof of work*: a new block verification process to find a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and which satisfies certain requirements. There are other alternatives to proof of work (eg. proof of stake) which are not related to this paper.

- *Consensus*: all or majority of participants verify new blocks. The valid block enters blockchain.

- *Incentive*: reward that encourages participants to contribute and stay honest.

The value of blockchain comes from its distributed consensus model. This model generates a trusted database in an untrusted community. Although participants may or may not trust each other, they all trust the transactions on the blockchain.

2.2 Distributed training

Deep learning models are typically trained by three steps: feed-forward evaluation, back-propagation, and weight updates. Feed-forward evaluation calculates a model's output for each input. Back-propagation calculates gradients based on the true value. Parameter weights are then updated according to gradients. After weight update, this process is repeated until the entire training dataset has been processed. This is called a training epoch. At the end of a training epoch, the model prediction error is computed on a validation set. Typically, training continues for many epochs, reprocessing the training dataset each time, until the validation set error converges to a desired low value. The trained model is then evaluated on test data.

This training process takes a long time (hours to months). Dean et al. [13] and Chilimbi et al. [14] applied data parallelism distributed training to speed up this process with multiple nodes. The system architecture they employed is shown in Figure 2. This architecture allows multiple replicas of the same model to be trained in parallel on different partitions of the training dataset. All the model replicas share a common set of parameters that is stored on a global parameter server. For

the speed of operation, each model replica operates in parallel and asynchronously publishes gradients to and receives updated parameter weights from the parameter server. While these asynchronous updates result in inconsistencies in the shared model parameters, neural networks are a resilient learning architecture and this mechanism has been proven to be working.

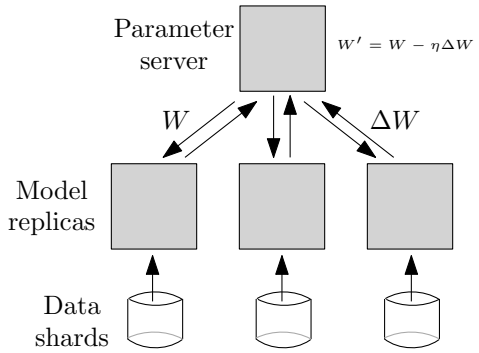


Figure 2: Distributed training system architecture

2.3 Related work

Hamm et al. [15], McMahan et al. [16] trained models from decentralized data with a central parameter server, which targets at intelligent behavior on mobile devices. They show that when two model replicas are trained independently with independent datasets and with the same initial state, naive parameter averaging achieves much lower loss than single dataset training. Smith et al. [17] present a distributed multi-task learning framework, taking communication cost, stragglers, and fault tolerance into consideration.

Shokri et al. [18] proposed a privacy-preserving deep learning system, which allows participants to train independently on their own datasets and selectively share small subsets of key parameters during training. On this basis, Le et al. [19] built an enhanced system via combining the privacy-preserving deep learning system with additively homomorphic encryption to prevent local data information from being leaked to the server. Bonawitz et al. [20] present a practical secure aggregation method for privacy-preserving machine learning with low communication and high robustness, which requires one server with limited trust as a message router and final result computing center among other parties.

All previous work relies on a trusted central server to do parameter exchange. BDML removes the need for this trusted central server, introduces more uncertainty and randomization into the scenario with a completely decentralized architecture. Furthermore, the BDML protocol encourages participants to contribute and stay honest, which is still a concern in the previous work.

3. BDML ARCHITECTURE

The distributed consensus model of blockchain has motivated us to create BDML. BDML's goal is to enable collaborative model training and evolution without exposing participants' private dataset.

3.1 A general architecture

Figure 3 shows BDML's blockchain design. Each block contains a model definition M and parameter weight W. The first block defines the optimization goal of this chain and constraint. For example, the optimizer could be the lowest loss and the constraint could be model size not exceeding 200MB. All participants of a BDML blockchain are expected to work on the same problem, each with their own secret dataset. No centralized parameter server is required. BDML guarantees that for a specific problem, the tail block always has the best model. The consortium collaboratively trains one model while keeping every participant's dataset unexposed.

Three types of transactions are defined:

1. Updated candidate weight W^c.
2. New candidate model structure definition $\{M^c\}$ and candidate weight $\{W^c\}$. This transaction is issued whenever a revolutionary model structure is proposed by a participant. This transaction is uncommon, as most new blocks are expected to be evolutionary weight updates.
3. Vote V. When $\{M^c, W^c\}$ or W^c is received, each participant runs local test set. If progress is made toward the optimization goal and constraints are met, a positive V is sent back. Otherwise, a negative V is sent back. In the case of negative V for $\{M^c, W^c\}$, the validator can optionally train $\{M^c, W^c\}$ with their secret dataset and send back updated model weight together with the vote.

The proof of work is designed by generating a $\{M^c, W^c\}$ or W^c which can obtain consensus from the consortium. Since training a model is time consuming but verification of its improvement is relatively easy, it matches the characteristic of proof of work well.

Consensus can choose different rules, such as

- A strict rule requires positive V received from all other participants.
- A loose rule requires positive V from the majority of the participants.
- A mixed rule requires positive V from all board members, and from the majority of the other participants.

The purpose of loose/mixed consensus rule is to improve the convergence speed of the model. For example, there may be a few participants who are too busy or unwilling to vote or their data distribution is very different from others. If the loose/mixed rule is used, the model convergence will be less affected.

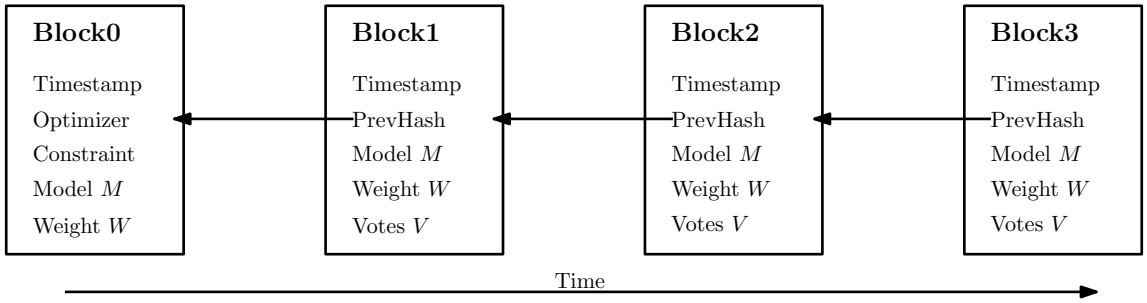


Figure 3: BDML blockchain

Once a participant successfully receives consensus from the consortium, M^c, W^c, and V are packed into a new block and broadcast out. The proof of work discussed above is a time-consuming task. In reality, only a few legitimate proposals ($\{M^c, W^c\}$ or W^c which can get consensus) will appear in the consortium over a period of time. These legitimate proposals will be broadcast out across the consortium, and participants who receive them will add them to the blockchain.

Reward coins are issued to the block creator by the system. The coins obtained by each block are not equal, which are related to many factors, such as the increase amount in accuracy and the number of participants. The hypothesis of BDML is that although data cannot be shared among all participants, the common goal of all participants is to train a better model. When a better model is trained, participants will get not only BDML coins, but also practical financial return. Taking e-healthcare as an example, a good tumor detection model is of great significance for medical diagnosis. The goal of this paper is to propose a collaborative training method. Incentive is a complex issue, we do not discuss specific reward strategies in this paper and leave it to future work.

The process to run the BDML blockchain is as follows:

1. Each participant pulls the latest M and W from the blockchain tail.

2. Each participant trains with their own dataset and stops training before overfitting, then broadcasts out $\{M^c, W^c\}$ or W^c.

3. Other participants vote upon receiving $\{M^c, W^c\}$ or W^c.

4. If consensus is reached, a new block is broadcast out and added to the blockchain. All other participants stop current training process and restart from the new block.

5. Before consensus is reached, each participant updates local parameters with all or selected parameter changes $\Delta W = W^c_{prev} - W^c$ from other participants and restarts from step 2.

The tail block always contains the best model for a specific problem. That said, as a distributed ledger, all historical blocks in the blockchain are available to the consortium. Participants have the option to retrieve knowledge from previous blocks.

3.2 Privacy concern

Traditional machine learning training requires a large amount of data. The goal of BDML is to solve the data privacy issue in collaborative training. Each participant keeps the data locally. The information transmitted across participants is the minimal update necessary to improve a particular model. Song et al. [21] demonstrated that a malicious machine learning algorithm can extract subsets of the training data and suggest that data holders should inspect the algorithm. In BDML the model structure is available on the blockchain and can be inspected by all participants. It's better for the training code to be open source within the consortium or be retrieved from a trusted provider. Hitaj et al. [22] devised a GAN based attack which is able to generate prototypical samples which look like the targeted training set but are not the real samples. Whether this is a privacy violation is controversial.

In BDML, each participant deposits some money and receives some initial blockchain coins when joining the consortium. Successful block creators also earn reward coins. When the model is good enough for commercial use, the generated profit can be shared by the whole consortium according to everyone's coin share. The deposit and incentive may help encouraging participants to play by the rule and stay honest. Krum [23] is a Byzantine-resilient algorithm for distributed SGD and can be employed in BDML to detect and preclude attackers. If an attacker is identified by the whole consortium, the attacker's share of coins can be forfeited. Those attackers ought to find it more profitable to play by the rule. In general, participants who send out valid W^c more frequently, who have more training data, who have more computation power, or who have better algorithm capability have more chance to win the reward.

3.3 A reference design

3.3.1 Overview

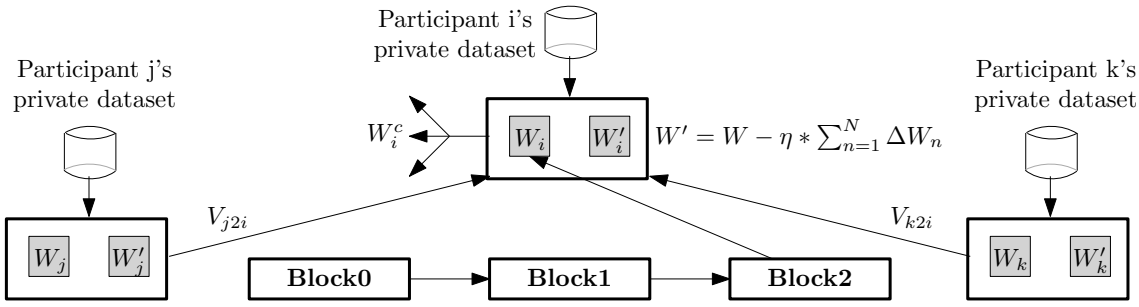


Figure 4: BDML decentralized training system architecture

Figure 4 shows the BDML training system architecture for a simple scenario.

We consider a set of n participants $\{P_i\}_{i=1}^n$. Each participant P_i has a local private dataset $S_i = \{(x_i^j, y_i^j)\}_{j=1}^{|S_i|}$. All participants in the blockchain consortium have a common learning objective. They keep using the same model structure and only update weight. The objective is formulated as

$$\min_{W \in \mathbb{R}} \mathcal{L}(W), \text{where } \mathcal{L}(W) = \sum_{i=1}^{n} \mathcal{L}_i(W) \tag{1}$$

which covers a large variety of models ranging from simple linear regression to deep learning. $\mathcal{L}_i(W)$ indicates the loss on participant P_i's local training set with parameters W. The common learning objective results in a global model for all participants, such as a classification model minimizing the prediction error rate over the union of all participants' dataset. Only two types of transactions are employed in this scenario: W^c (updated candidate weight) and V (vote).

An IID (independent and identically distributed) data distribution is assumed and the local parameter update algorithm is designed for IID. Given an unknown and fixed distribution μ over $\mathcal{X} \times \mathcal{Y}$, each participant P_i's local dataset S_i is drawn from μ. The size of local dataset for each participant can be different. The local dataset S_i is split into the training set $S_i^{train} = \{(x_i^j, y_i^j)\}_{j=1}^{|S_i^{train}|}$ and the test set $S_i^{test} = \{(x_i^j, y_i^j)\}_{j=1}^{|S_i^{test}|}$. For non-IID data distribution, a more advanced local parameter update algorithm may be required and meta-learning [24] technology may be needed. As BDML is a general architecture and the local parameter update algorithm can be designed for specific scenarios, we only do experiments with the IID scenario and leave the non-IID scenario to future work.

One of the core components of the BDML system is a parameter sharing protocol, which enables each participant P_i to converge to a local model by minimizing the loss over their local private training set S_i^{train} independently and lets participants to improve upon their local models by leveraging information from other participants in the blockchain consortium to learn a shared global model. We assume that each participant P_i maintains a local neural network parameter W_i for local training and a local W_i' for receiving knowledge from other participants.

The weight in the first block is randomly generated, that is, all participants starts from the same random initialization. Each participant P_i initializes the parameters according to the first block and then runs Algorithm 1 (training and collecting votes) and Algorithm 2 (voting and receiving knowledge) on their private training/test set in parallel repeatedly and competes with other participants to generate a new block.

3.3.2 Training and collecting votes

Algorithm 1 provides the pseudocode for the process of training and collecting votes.

Algorithm 1 Training and collecting votes

$\{W_i, W_i'\} \longleftarrow W_{tail}$
while no new block is found **do**
  while not converge **do**
    train W_i with P_i's private training set S_i^{train}
  end while
  $W_i^c \longleftarrow W_i$
  test W_i^c with P_i's private test set S_i^{test}
  while $acc(W_i^c) > acc(W_{tail})$ **do**
    broadcast candidate weight W_i^c
  end while
  wait for V from other participants
  if all V are positive **then**
    create a new block with W_i^c and all votes
    add the new block to blockchain by broadcasting
  else
    update W_i with W_i'
  end if
end while

When a new block is generated, the participant P_i pulls the parameters W_{tail} from the tail of the blockchain and overwrites local parameters W_i and W_i' with the pulled values. Then the participant P_i trains the local model W_i by minimizing the loss on local training set

S_i^{train} over many epochs:

$$W_i \in \arg\min_{W \in \mathbb{R}} \mathcal{L}_i(W) \tag{2}$$

The training process can be done with a sequence of mini-batches. A mini-batch is a set of examples randomly chosen from the training set at a given batch size. If stochastic gradient descent (SGD) is applied, the update rule for a parameter W_i is

$$W_i = W_i - \alpha \frac{\partial \mathcal{L}_i(W_i)}{\partial W_i} \tag{3}$$

where α is the learning rate.

Participants do not need to communicate with each other during their local training. They influence each other's training via the parameter sharing protocol. When the participant P_i independently converges to a local model W_i on training set S_i^{train}, the participant P_i assigns the values of W_i to W_i^c as a candidate weight, then evaluates W_i^c with local test set S_i^{test}. The baseline W_{tail} is the model on the tail of the blockchain. When $acc(W_i^c) > acc(W_{tail})$, which means the accuracy of W_i^c is better than the baseline, the participant P_i broadcasts the candidate weight W_i^c and waits for votes from other participants.

Consensus can choose different rules. In Algorithm 1, we take the strict rule as an example. According to the strict rule, when all votes are positive, a new block is generated and added to the blockchain by broadcasting. If the consensus is not reached, the participant P_i updates W_i with W_i':

$$W_i = \beta W_i + (1-\beta) W_i' \tag{4}$$

where β can be set according to various factors, such as the size of the dataset, the number of participants, the number of local training epochs. To avoid participants drifting too far from the global optima and getting difficult to reach consensus, each participant take a certain probability to revert their local training model parameters back to W_{tail}. As long as no new block is generated, the probability increases as time passes away.

3.3.3 Voting and receiving knowledge

Algorithm 2 provides the pseudocode for the process of voting and receiving knowledge.

When the participant P_i receives W_j^c from someone else, P_i evaluates W_j^c with local test set S_i^{test}. If $acc(W_j^c) > acc(W_{tail})$, which means the accuracy of W_j^c is better than the current baseline, P_i sends back a positive vote. Otherwise, P_i sends back a negative vote.

At the same time of voting, P_i maintains a local vector of neural network parameters W_i' to receive knowledge from other participants by accumulating others' gradients.

$$W_i' = W_i' - \eta * (W_{j.prev}^c - W_j^c) \tag{5}$$

The gradient $g_j = W_{j.prev}^c - W_j^c$ is the change between the last and current candidate weights from the participant P_j. The term gradient typically refers to the change in a parameter after training over a mini-batch, we generalize the term gradient to one or more epochs of training here.

Algorithm 2 Voting and receiving knowledge

```
while no new block is found do
  if W_j^c received then
    W_i' = W_i' − η ∗ (W_{j.prev}^c − W_j^c)
    test W_j^c with P_i's private test set S_i^test
    if acc(W_j^c) > acc(W_tail) then
      broadcast positive vote V_{i2j}
    else
      broadcast negative vote V_{i2j}
    end if
  end if
end while
```

Recent works [16,25–27] indicate that when two models start from the same initialization and then train independently on different subsets of the data, parameter averaging works well for non-convex objectives. As shown in section 3.3.2, all BDML participants start from the same random initialization in the first block, and overwrite their locally learned parameters with the new block once a new block is generated, which effectively update all participants' models with a new common initialization state. Updating W_i with W_i' can indirectly learn characteristics from other participants' private training result via the parameter sharing protocol. With the generation of new blocks, BDML can obtain model convergence and achieve better accuracy than any single participant can achieve alone.

4. EXPERIMENTAL RESULTS

4.1 Experimental setup

We implement the BDML reference design described in section 3.3 with P participants and design our own blockchain for BDML scenario. The system implementation is based on an open source project [28]. For the training and vote collecting process, each participant independently trains a model replica for at least T epochs with their private dataset before broadcasting W^c. To make the experiment deterministic and repeatable, we design a dedicated voting stage for each round when all participants finish T epochs. In each voting stage, N out of P participants are allowed to broadcast W^c and collect votes.

Image classification tasks are adopted to explore BDML's feasibility and applicability. We evaluate BDML on three popular datasets:

- *MNIST* [1], which is a dataset of handwritten digits for digits recognition task. Each image in the MNIST dataset is a grayscale 28 × 28 pixel image. We use 60,000 images as the training set, and 10,000 images as the test set.

- *CIFAR-10* [2], which has 60,000 32×32 color images of 10 classes, with 6,000 images per class. We use 50,000 images as the training set, and 10,000 images as the test set.

- *LiTS* [3], which is the dataset of Liver Tumor Segmentation Challenge containing 131 contrast-enhanced 3D abdominal CT scans. We use 87 scans as the training set, and 44 scans as the test set.

MNIST and CIFAR are relatively small data sets, while LiTS is a large medical data set. The reason for using these data sets is to validate the adaptability of BDML to different data types and data sizes.

We use three popular deep learning models:

- *LeNet-5* [9], which is a convolutional neural network comprising 7 layers (conv $\rightarrow$ pool $\rightarrow$ conv $\rightarrow$ pool $\rightarrow$ fully connected $\rightarrow$ fully connected $\rightarrow$ output). We do extensive experiments with the LeNet-5 model on the MNIST dataset to evaluate the impact of numbers of participants, consensus rules, dataset distribution, etc.

- *VGG-16* [5], which is a 16-layer deep network containing multiple 3×3 convolution and 2×2 pooling layers. To demonstrate BDML's applicability on more complex tasks, we evaluate BDML with the VGG-16 model for the CIFAR-10 image classification task. Dropout and batch normalization are applied to prevent overfitting [29].

- *Unet* [10], which consists of a contracting path (left side) and an expansive path (right side). In total the network has 23 convolutional layers. We evaluate BDML with the Unet model for the LiTS liver segmentation task.

Two dataset distributions are studied:

- *Even*, where the dataset is shuffled and then evenly partitioned to each participant;

- *Poisson*, where the dataset is shuffled and then partitioned to each participant according to the Poisson distribution, which is more in line with the real application scenarios.

Two consensus rules are studied:

- *Strict*, which requires positive votes from all other participants;

- *Loose*, which requires 80% positive votes from other participants.

4.2 MNIST/LeNet-5 result

4.2.1 *Number of participants*

We first experiment with the number of participants P. MNIST dataset is shuffled and evenly distributed to P participants as their private dataset. Each participant trains the LeNet-5 model with their private dataset for 20 epochs. Then a random number of participants broadcast W^c for voting. The *strict* consensus rule is adopted, which requires positive votes from all other participants.

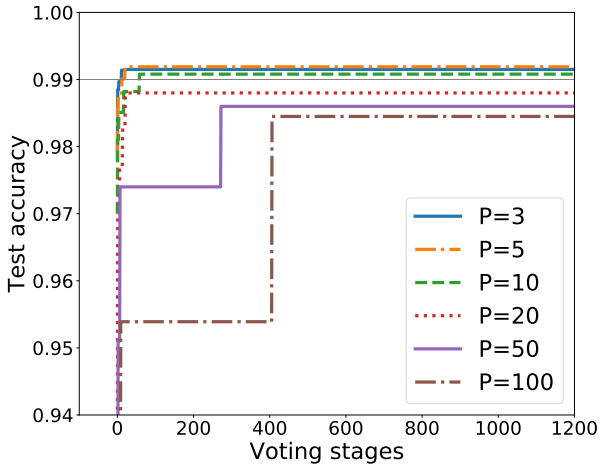


Figure 5: Accuracy vs. participant number

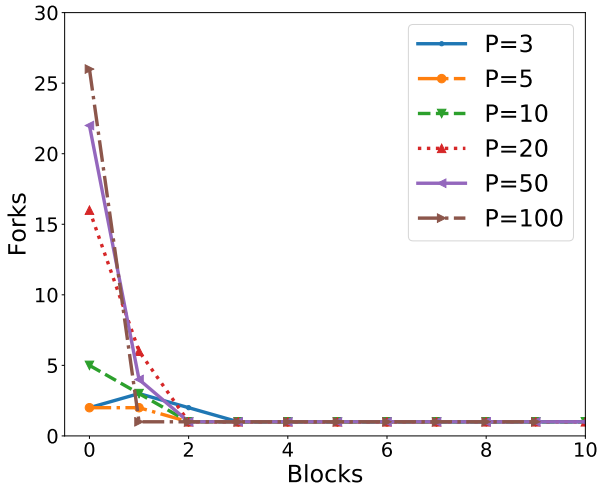


Figure 6: Accidental fork vs. participant number

Figure 5 shows the test accuracy of the tail block for different numbers of participants. With more participant number, it becomes more difficult to reach consensus and BDML converges slower. For this simple MNIST digit recognition task with the LeNet-5 model, $P \leq 10$ is recommended.

It is possible that more than one participants broadcast W^c at the same time and reach consensus. In this

case, an accidental fork happens. There are two solutions for accidental forks. The first solution is to wait until subsequent blocks are added and one of the chains becomes longer than alternatives. The longest chain will be the only valid chain and all participants eventually converge on this chain. The other solution is to employ the PBFT [30] algorithm and utilize a primary to serialize the request to avoid forks. During our experiment, as our goal is to find out the frequency of forks, we just randomly select one fork to be the valid chain and ignore the others when an accidental fork occurs.

Figure 6 shows the number of forks generated during the training process for different numbers of participants. Experiments show that accidental fork only happens at the first few blocks. After that, it is uncommon to have accidental forks.

4.2.2 BDML accuracy

To evaluate whether the accuracy of the models produced by BDML is close to the centralized, privacy-violating case where the model is training on the union of all participants' dataset, and whether BDML can achieve better accuracy than any single participant can achieve alone; three experiments are performed:

1. *Super_Dataset*: Train the LeNet-5 model with the full MNIST dataset;
2. *BDML*: Train the LeNet-5 model with BDML ($P = 10, T = 20, N = random$, *even*, *strict*);
3. *Sub_Dataset*: Shuffle the MNIST dataset and distribute to 10 participants evenly. Then each participant trains the LeNet-5 model with private dataset without communication. Each participant's test accuracy is evaluated on the full MNIST test set.

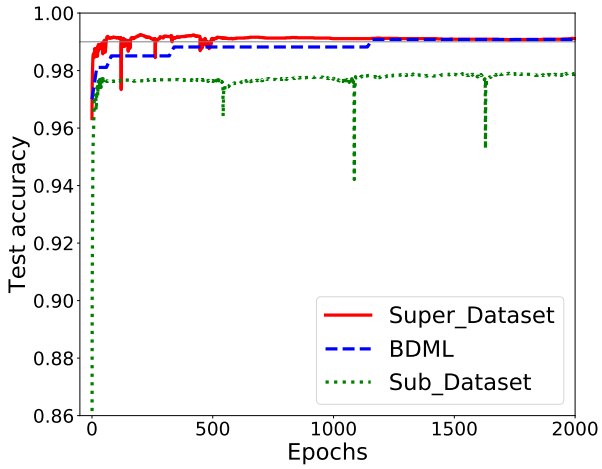


Figure 7: BDML accuracy

Figure 7 shows the results of the above three experiments. Since each participant converges to similar accuracy in the *Sub_Dataset* experiment, we only show the accuracy curve for one of the participants.

The comparison of the accuracy curve of BDML and the *Super_Dataset* experiment shows that BDML can achieve *Super_Dataset* accuracy, which implies that the accuracy of the LeNet-5 model produced by BDML is close to the centralized, privacy-violating case. As shown in Figure 7, the accuracy of BDML far outperforms the accuracy in the *Sub_Dataset* experiment, which indicates that BDML can achieve better accuracy than any single participant can achieve alone. In situations where super dataset training is not possible, BDML provides a way to achieve super dataset training accuracy.

4.2.3 Consensus

To evaluate the impact of the consensus rule on BDML, we compare the accuracy of the *strict* rule and the *loose* rule. Figure 8 shows the test accuracy for the *strict* rule and the 80% *loose* rule ($P = 10, T = 20, N = random$, *even*). The *loose* rule generates a new block if 80% of participants send back positive votes.

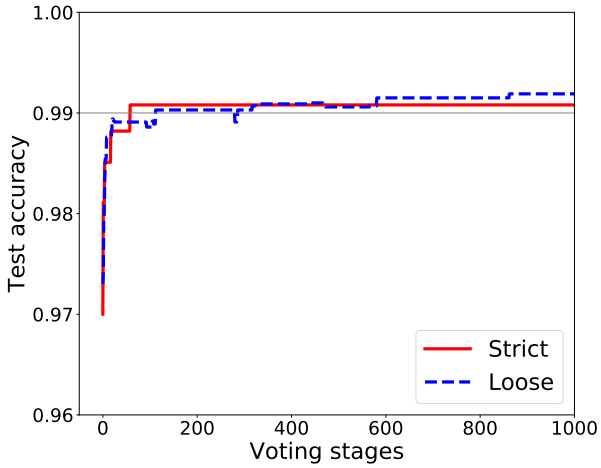


Figure 8: Consensus rule

As shown in Figure 8, the *loose* consensus rule reduces the difficulty to reach consensus. As a result, the *loose* rule suffers from accidental test accuracy decrease which is not possible with the *strict* rule. However, the *loose* rule eventually achieves higher accuracy. Each of the two rules has its own advantages and disadvantages. In practice, consensus rules can be selected according to needs. And hybrid methods are also possible, which employs *strict* rule in the early stages and switches to *loose* rule in later stages.

4.2.4 Broadcasting frequency

Figure 9 shows that more broadcasting participants N in each voting rounds is preferred for better accuracy and faster convergence ($P = 10, T = 20$, *even*, *strict*). However, broadcasting has high communication cost. It is desirable to reduce this cost. One way to do this is to increase the training period T. Figure 10 shows that a relatively big T (20, 50) works fine. But a too

large T (100, 200) reduces the test accuracy apparently ($P = 10, N = random$, *even*, *strict*).

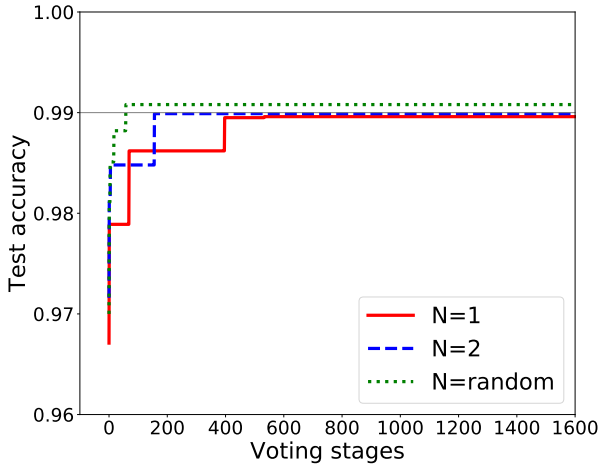


Figure 9: Impact of broadcasting participants

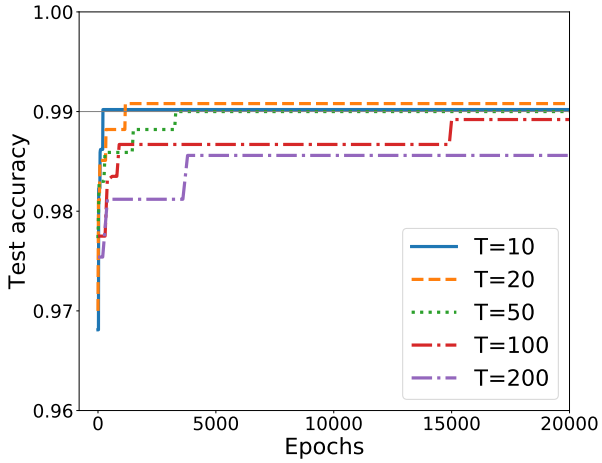


Figure 10: Impact of training period

4.2.5 *Dataset distribution*

We study two methods to distribute the MNIST dataset, which has 60,000 images. With 10 participants, the *even* method distributes 6,000 images to each participant. The *Poisson* method first distributes 3,000 images to each participant to ensure that each participant has a minimum number of training samples. Then the rest 30,000 images are distributed to all participants according to the Poison distribution. The *Poisson* method is more common in real scenarios.

Figure 11 shows the test accuracy for those two methods, which doesn't exhibit an apparent difference ($P = 10, T = 20, N = random$, *strict*).

4.3 CIFAR-10/VGG-16 result

To demonstrate BDML's applicability on more complex tasks, we evaluate BDML with the CIFAR-10 image classification task.

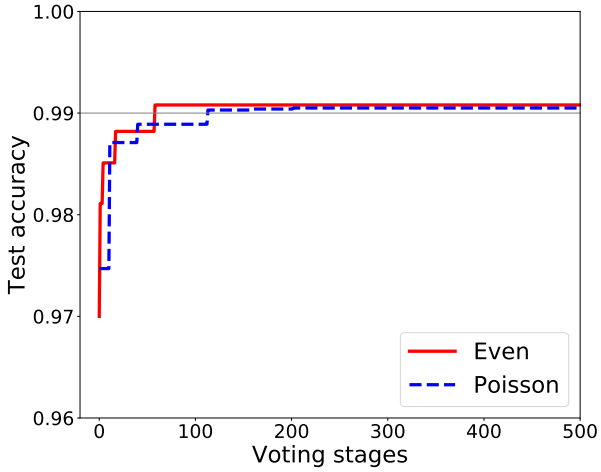


Figure 11: Even vs. Poisson

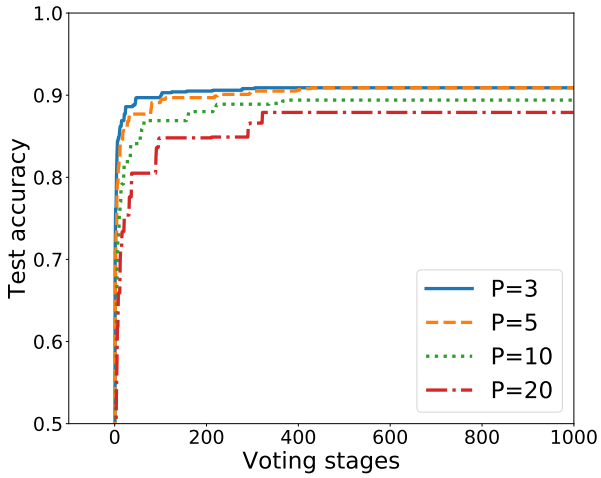


Figure 12: Accuracy vs. participant number

Figure 12 shows the test accuracy for different numbers of participants ($T = 10, N = random$, *even*, *strict*). With more participant number, it becomes more difficult to reach consensus and BDML converges slower. Figure 13 shows the impact of broadcasting frequency ($P = 10, N = random$, *even*, *strict*). A suitable T can reduce the communication cost, and make the model converge in a reasonable time. Figure 14 shows that the accuracy of the model produced by BDML ($P = 10, T = 5, N = random$, *even*, *strict*) is close to *Super_Dataset* accuracy and outperforms *Sub_Dataset* accuracy.

As discussed above, the CIFAR-10 image classification task shows similar behavior as experiments on the MNIST dataset. The experimental results show that for complex tasks, the accuracy of the models produced by BDML is still close to the super dataset training and can achieve better accuracy than any single participant

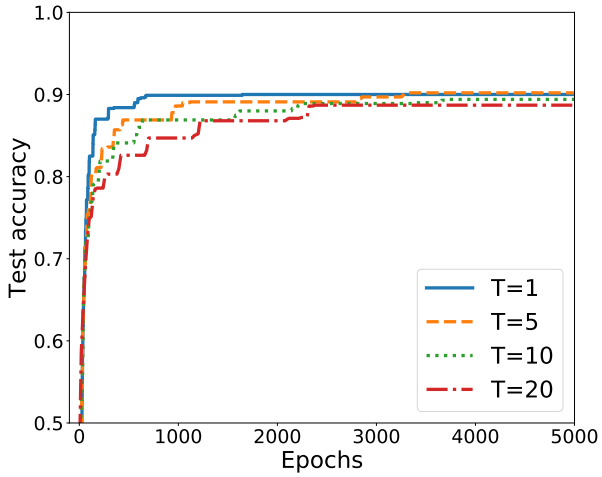


Figure 13: Impact of training period

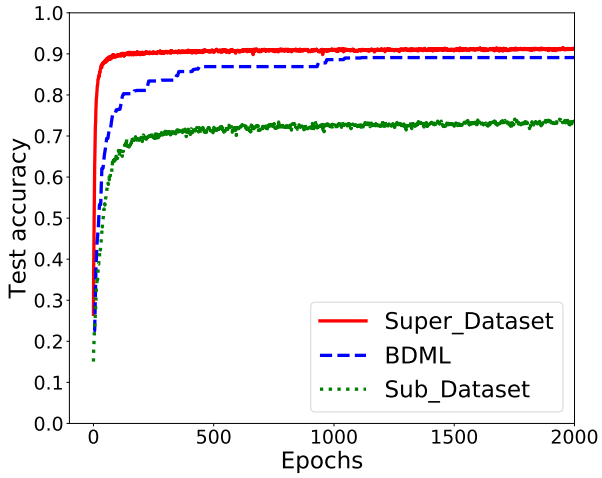


Figure 14: BDML accuracy

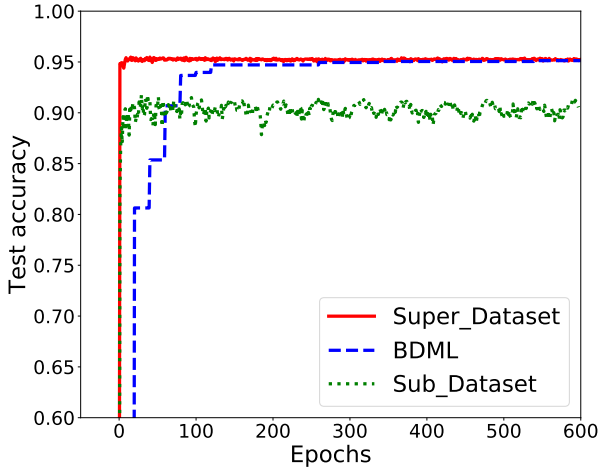


Figure 15: BDML accuracy

can achieve alone.

4.4 LiTS/Unet result

E-healthcare is a typical application scenario of BDML. We evaluate BDML with the LiTS liver segmentation task.

Figure 15 shows that the accuracy of the model produced by BDML ($P = 10, T = 20, N = random$, *even*, *loose* (80%)) is close to *Super_Dataset* accuracy and outperforms *Sub_Dataset* accuracy.

4.5 Storage space analysis

A BDML block typically contains a set of parameters, the required storage space varies from a few MB (e.g., relatively simple networks such as LeNet-5) to several hundred MB (e.g., complex networks such as VGG-16). The model usually needs more than a dozen blocks to converge, so the storage space required for a blockchain varies from several tens of MB to several GB.

5. BDML APPLICATIONS

As a collaborative model training framework, BDML is best suited for data privacy sensitive scenarios, especially when more than one organizations are involved. For example, clinical data in hospitals are very sensitive and not supposed to be shared. Therefore, e-healthcare researchers can only train models on datasets belonging to their own institutions. It is obvious that when the training dataset becomes larger and more diverse, the trained model becomes better. E-healthcare researchers can utilize BDML to collaborate with each other and develop better models to diagnose diseases. BDML is particularly useful for the rare disease scenario where any single institution has too few data samples for training. In addition, bank financial analysis, enterprise supply chain management, etc. are some other possible applications.

6. CONCLUSION AND FUTURE WORK

In this paper, we present a blockchain-based distributed machine learning architecture (BDML) allowing researchers in a consortium to share implicit knowledge about their data without exposing the data, which targets at consortium blockchains and resolves the data privacy concern and ensures trusted collaborative model training. We evaluate BDML's feasibility and applicability on image classification tasks and the e-healthcare scenario. The experimental results show that BDML is practical and can achieve super dataset training accuracy.

There are still a lot of challenges ahead. Technologies to scale participant number to a larger scale and reduce communication cost are interesting directions for future work. Parameter update algorithms for non-IID and heterogeneous data scenarios, reward policy, and applying BDML to public blockchains are also interesting future research directions.

7. REFERENCES

[1] Yann LeCun. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 2018.

[2] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[3] Lits - liver tumor segmentation challenge. http://www.lits-challenge.com/, 2017.

[4] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR Workshop*, 2016.

[5] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 15)*, 2015.

[6] BVLC. Model zoo. https://github.com/BVLC/caffe/wiki/Model-Zoo, July 2018.

[7] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[8] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283, 2016.

[9] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[10] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[11] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008.

[12] Michael Crosby, Nachiappan, Pradhan Pattanayak, Sanjeev Verma, and Vignesh Kalyanaraman. Blockchain technology beyond Bitcoin. Technical report, 2015.

[13] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Marc'aurelio Ranzato, Andrew Senior, Paul Tucker, Ke Yang, Quoc V. Le, and Andrew Y. Ng. Large scale distributed deep networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1223–1231. Curran Associates, Inc., 2012.

[14] Trishul Chilimbi, Yutaka Suzue, Johnson Apacible, and Karthik Kalyanaraman. Project adam: Building an efficient and scalable deep learning training system. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 571–582, Broomfield, CO, 2014. USENIX Association.

[15] Jihun Hamm, Jackson Luken, and Yani Xie. Crowd-ml: A library for privacy-preserving machine learning on smart devices. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6394–6398, 2017.

[16] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[17] Virginia Smith, Chao Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. Federated multi-task learning. 2017.

[18] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321. ACM, 2015.

[19] Trieu Phong Le, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics & Security*, PP(99):1–1, 2018.

[20] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan Mcmahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *ACM Sigsac Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[21] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, pages 587–601, New York, NY, USA, 2017. ACM.

[22] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and*

Communications Security, CCS '17, pages 603–618, New York, NY, USA, 2017. ACM.

[23] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems 30*, pages 119–129. Curran Associates, Inc., 2017.

[24] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[25] Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2933–2941, 2014.

[26] Ian J. Goodfellow, Oriol Vinyals, and Andrew M. Saxe. Qualitatively characterizing neural network optimization problems. *Proceedings of the 3rd International Conference on Learning Representations (ICLR 15)*, 2015.

[27] Anna Choromanska, Mikael Henaff, Michaël Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, pages 192–204, 2015.

[28] dvf. A simple blockchain in python. https://github.com/dvf/blockchain, 2018.

[29] Shuying Liu and Weihong Deng. Very deep convolutional neural network based image classification using small training sample size. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, pages 730–734. IEEE, 2015.

[30] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, OSDI '99, pages 173–186, Berkeley, CA, USA, 1999. USENIX Association.

Blockchain Based Solution to Improve Supply Chain Collaboration

Feifei Chen
Lenovo Research
Lenovo HQ East, Beijing
+86-18010195190
chenff3@lenovo.com

Qingxiao Guo
Lenovo Research
Lenovo HQ East, Beijing
+86-18701099715
guoqx2@lenovo.com

Xiaobing Guo
Lenovo Research
Lenovo HQ East, Beijing
+86-18519565781
guoxba@lenovo.com

Ajay Dholakia
Lenovo Data Center Group
7001 Development Drive, Morrisville
+1-919-237-8118
adholakia@lenovo.com

Yi Zheng
Lenovo Data Center Group
Lenovo HQ West, Beijing
+86-13910730264
zhengyi5@lenovo.com

Jierong Dong
Lenovo Data Center Group
Lenovo HQ West, Beijing
+86-18519555325
dongjr1@lenovo.com

Guiping Zhang
Lenovo BT/IT
Lenovo HQ East, Beijing
+86-13651189520
zhanggpb@lenovo.com

Jingsheng Li
Lenovo BT/IT
Lenovo HQ East, Beijing
+86-18911775087
lijs6@lenovo.com

Yunhao Wang
Lenovo Research
Lenovo HQ East, Beijing
+86-18911776005
wangyh43@lenovo.com

ABSTRACT

In contemporary society, most companies have to employ long and diverse supply chain networks for raw materials purchasing or product sales and distribution. An efficient and collaborative supply chain network is essential for enterprise operation. Therefore, in the supply chain network, not only the core enterprise but also all the participants' operating efficiencies need to be considered. Moreover, in traditional supply chain network management, there are several challenges such as lack of data transparency and data inconsistency for all participants, which leads to low collaboration efficiency for the supply chain network. This paper will present a blockchain based solution to ameliorate these problems and improve supply chain collaboration. This new solution is validated in a real business process at Lenovo and has delivered better performance.

Keywords

Supply Chain Transparency, Supply Chain Collaboration, Blockchain.

Second International Symposium on Foundations and Applications of Blockchain (FAB '19)
April 5, 2019, Los Angeles, California, USA.

1. INTRODUCTION

In November 2008, Satoshi Nakamoto introduced the basic principles of Bitcoin and the basic concepts of blockchain [1]. Blockchain, as the underlying technology of Bitcoin, is a type of distributed computing paradigm, sequentially connecting data blocks in chronological order that constitutes a chained data structure and uses cryptographic techniques to implement consistent storage of data with protection against tampering and repudiation. As a new type of computing paradigm and cooperating mode whose trust is built with lower cost in untrusted competitive environment, blockchain has been changing application scenarios and operating rules in many industries, and it is highly valued in the areas of digital currency, financial service, Internet of Things, intelligent manufacturing, medical health, credit reporting, and so on [2]. At the same time, blockchain features of shared ledger and tamper-proof data have laid the foundation for its application in supply chain.

A supply chain is a complex functional network consisting of suppliers, manufacturers, distributors, retailers and consumers, enabling enterprises to handle the flow from raw materials to finished goods. The participants in a supply chain are expected to collaborate with each other and build relationships and create a foundation for trust, thereby achieving overall efficiency optimization and improvement through inter-enterprise collaboration. This, in turn, is expected to bring greater value and benefits to all the

participants in the supply chain. In this network, commodity flow, logistics, information flow, and capital are intertwined, and the coordination difficulty is extremely high [3]. Information is distributed across different enterprises in the supply chain. The degree of information sharing and the running speed are slow, and the information authenticity and reliability are poor [4]. This creates the so-called "information island" phenomenon in current supply chain operation. As a kind of decentralization technology, blockchain has the characteristics of distributed processing, with attributes such as shared storage and network-wide consensus confirmation providing a good solution to the problems of data storage insecurity and sharing difficulties in traditional centralized systems. Thus, blockchain as an emerging technology is gradually finding applicability in the supply chain domain.

In August 2016, Bank of America, HSBC and the Singapore government established a blockchain letter of credit trade for import and export of commodities project based on the Hyperledger project [5]. In October 2016, Wal-Mart, IBM and Tsinghua University jointly created a blockchain-based industrial supply chain project. The goal is to make supply chain data more accurate and secure [6]. In March 2017, IBM and Maersk worked together to build a blockchain solution based on Hyperledger Fabric. IBM and Maersk intend to work with a network of shippers, freight forwarders, ocean carriers, ports and customs authorities to build the new global trade digitization product [7].

In this paper, we propose a blockchain based solution to improve sharing in and ensure the authenticity and security of data in a supply chain system, and promote synergy between participating enterprises. Furthermore, a key aspect of the solution described here is data sharing enabled by enhanced privacy protection using cryptographic techniques that are incorporated over and above the capabilities available in the basic Hyperledger Fabric platform.

The rest of this paper is organized as follows. Section 2 provides an overview of blockchain technology. Section 3 describes key challenges currently encountered in supply chain management scenarios. Section 4 is devoted to the solution being developed and deployed by Lenovo to address the aforementioned challenges. This section details our experiences in designing and implementing the solution. Finally, Section 5 summarizes our conclusions so far and also identifies areas of ongoing and future research and development work.

2. BLOCKCHAIN TECHNOLOGY

Blockchain technology is a specific realization of Distributed Ledger Technology (DLT). It includes four key technologies: Awarding, Block-chain storage, Consensus, and Decentralization. We can call this the ABCD of blockchain.

The purpose of the Awarding mechanism is to encourage participants to continue to invest resources for the long-term running and stability of the blockchain network. This is typical for the so-called public or permissionless blockchain networks. However, in blockchain enterprise application such as enterprise consortium blockchain, it is not a necessary feature. The award mechanism is implicit because all the participants can get benefits by business collaboration in the consortium network instead of an explicit token incentive. This is typical for the so-called private or permissioned blockchain networks.

Block-chain storage refers to the data structure of blockchain. A block is the smallest unit for data storage and chain is the link of blocks using some hash algorithms. This structure ensures data integrity and traceability of information stored in the blockchain.

Consensus is the core of blockchain, which is used to ensure data consistency among the peers in a blockchain network. Common consensus algorithms are POW (Proof Of Work), POS (Proof Of Stake), DPOS (Delegated Proof Of Stake), BFT (Byzantine Fault Tolerance), RAFT etc. POW, POS, and DPOS are mostly used in public blockchain for leader peer selection and BFT or RAFT is usually used in the consortium or private blockchain to keep eventual consistency of distributed ledgers. Consensus combined with block-chain storage ensures data immutability. It's another important feature for blockchain that can be a trust data sharing platform.

Decentralization is the fourth key attribute of a blockchain, referring to the topology of system implementation. Decentralization does not mean complete absence of center. We regard multi-center or weak center as decentralization. Decentralization helps blockchain to have the characteristics of anti-single point of attack and high fault tolerance.

3. SUPPLY CHAIN CHALLENGES

Supply chain involves an end-to-end business process from raw material procurement, manufacturing and warehouse storage of product to its sale and distribution. Supply chain management is the discipline aimed at examining and managing supply chain networks with the goals of achieving cost savings and better customer service. An important objective is to improve the competitiveness of an enterprise in global markets in spite of strong competitive forces and rapidly changing customer needs [8]. Many articles in connection with the theory and practice of Supply Chain Management (SCM) have been reported over the period of last 20 years, but the subject matter is still under important improvement and discussion [9]. We point out three major challenges in supply chain management here: lack of transparency, inconsistent data used by all participants in a supply chain network and low efficiency of multi-party collaboration.

3.1 Lack of Transparency

One of the biggest obstacles to efficient supply chains is the lack of transparency and the inability to respond quickly. Awaysheh & Klassen [10] identify transparency as the extent to which information is readily available to both counterparties in an exchange and also to outside observers. IBM pointed out that 80% of the world's data is unstructured, but businesses are only able to gain visibility into a portion of that data [11]. 84% of Chief Supply Chain Officers (CSCOs) say the lack of supply chain visibility is their top concern [12] and 87% of CSCOs say it's difficult to predictand proactively manage disruptions [13]. Traditional IT systems like ERP were built to record supply chain data. However, supply chain is becoming more and more complex due to global sourcing and continuous trend of leaning down. A company has many stakeholders in its supply chain network such as suppliers, carriers, governments, brokers, customers and so on. Over the years, most companies did supply chain integration (SCI). The National Research Council [14] provides a comprehensive definition of SCI as "an association of customers and suppliers who, using management techniques, work together to optimize their collective performance in the creation, distribution, and support of an end product manufacturer". However, most SCI systems were implemented in traditional way and generate a cobweb-like supply chain network (see Figure 1).

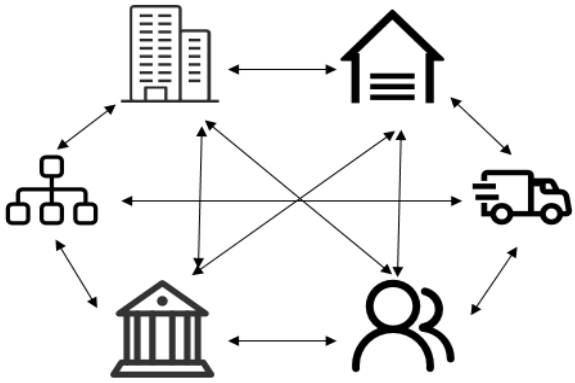


Figure 1. Traditional supply chain network.

As we can see in the Figure 1, although a lot of point-to-point systems integration work may have been done, it is still difficult for all participants to obtain all the data in the network. Thus, for the whole process and for all participants, lack of transparency is still the biggest challenge.

3.2 Data Inconsistency

As shown in Figure 1, in a traditional supply chain network with point-to-point integration, each participant the network has only a part of the information and stores the data in their own system. Furthermore, each organization can change their own data freely, which may result in inconsistent data across multiple participants. In order to ensure the benefits of an efficient supply chain, enterprises must share information, requiring the construction of a core data. Such a unified enterprise data center platform will effectively improve the sharing and utilization of data, provide support for enterprise management and decision-making, and enable supply chain participants to quickly respond to external changes. However, the creation of enterprise data centers faces the data sharing problem among business sectors and partners. Enterprise business systems such as ERP systems, procurement systems, IT Service Management systems operate independently and these independent, heterogeneous, closed systems form the "information isolated island", limiting the efficiency of the core business [15].

3.3 Low Collaboration Efficiency

Collaboration and co-operation within the company organizations and trading partners are important for truly removing waste across the entire supply chain. Accelerating cycle time, increasing inventory velocity and reducing costs for the high-volume and high-margin products can affect return on investment and drive the benefit of lean for everyone to see [16]. As we mentioned before, today's supply chain is a complex, multi-party, long chain network. As Lei Wen points out [17] the entire chain connects customers, retailers, distributors, manufacturers and suppliers, beginning with the creation of raw material or component parts by suppliers and ending with consumption of the product by customers. Information collaboration is the foundation of Supply chain collaboration which is one of the critical activities for firms to gain competitive advantage and achieve the business objectives of the whole supply chain.

4. BLOCKCHAIN SOLUTION

4.1 Lenovo Buy & Sell Business Case

Here we present a real Lenovo business case in Lenovo raw material trade process. There are three participants in this "Buy & Sell" scenario: ODM (Original Design Manufacturer), Lenovo and Supplier. In this case, some Lenovo products are manufactured by ODM. Considering production cost and product quality, raw materials used in ODM manufacturing need to be purchased by Lenovo, and Lenovo purchases from qualified suppliers. This is a simple multi-party trading scenario. The business process is shown in Figure 2.

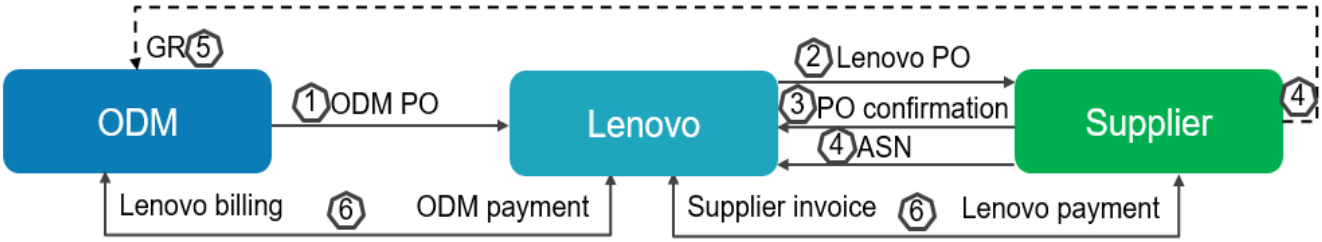


Figure 2. Lenovo Buy & Sell business process.

1) ODM sends raw materials purchase order (PO) to Lenovo.
2) Lenovo receives ODM PO then converts to Lenovo PO to supplier.
3) Supplier gets Lenovo PO and sends a response message called PO confirmation.
4) When supplier completes the stock-up, it sends shipment information to Lenovo (ASN) and physical goods to ODM directly.
5) ODM based on physical goods does goods receive (GR).
6) Lenovo bills to ODM, ODM pays, Lenovo receives supplier invoice and makes payment to supplier.

In this process, all the participants have some pain-points in using traditional point-to-point supply chain integration, along the lines described in the previous section. The main challenges are as follows:

- It is difficult for ODM to know whether the supplier accepts the order or not and the status of the order processing. Lack of transparency will affect the production planning of the ODM.
- Lenovo participates in the whole process of the transaction. The inconsistency between information flow and physical goods flow results in a large amount of manual information checking and communication.
- Suppliers is unable to obtain the actual demand from ODM in the first time, results in the compression of production and stocking cycle of supplier.
- Each participant's system outage will affect the whole process.

4.2 System Design Requirements

In order to solve the problems in the Buy & Sell use case described above, a new system design is needed for all participants and we list the important requirements as follows.

1) A new way of interaction is needed to increase data transparency for all participants, with which each one can share or get the data freely.
2) We need to ensure data consistency among participants. All transaction history data cannot be tampered with and all the data can be traceable.
3) A distributed system is needed to prevent a single point of failure from causing the overall process to fail.
4) We must ensure adequate privacy protection, although we need to increase the data transparency. Transparency and privacy are not entirely opposed as Wüst and Gervais mentioned in [18].

However, Wüst and Gervais [18] provide a flow chart to determine whether a blockchain is the appropriate technical solution to solve a problem. In their view, if all the participants in a network are trusted, they should not use blockchain. In fact, even if all the participants are trusted in the alliance, but if each participant stores data in their own system, financial settlement steps still require cross-system reconciliation, making trade friction inevitable. But using blockchain technology, all the ledger data is generated by smart contracts and no one can tamper with the data, allowing a reduction in trade friction.
Therefore, a permissioned blockchain with consensus, provenance, immutability and privacy protection appears to be a sensible solution.

4.3 System Solution

With the proposed blockchain based solution, business participants reform original system and remove the point-to-point multiparty system integration. ODM, Lenovo and supplier together use Lenovo blockchain platform to build an alliance network. Lenovo blockchain platform is based on Hyperledger Fabric and makes some enhancements. The most important enhancement is that we use cryptography to encrypt sensitive information on the chain. Only authorized participants in the trade process can obtain the information, while other participants only keep complete ciphertext data. We use same shared ledger for all participants to store transaction data and ensure that each party has appropriate data access authority. This means that all the participants' peers store the same data but only the transactions related parties have the private key for sensitive data decryption to get the detail information. We design and develop a smart

contract to record data in ledger, we also setup event hub in smart contract to listen to the ledger changes and catch the related event to automatically trigger some business process (see Figure 3).

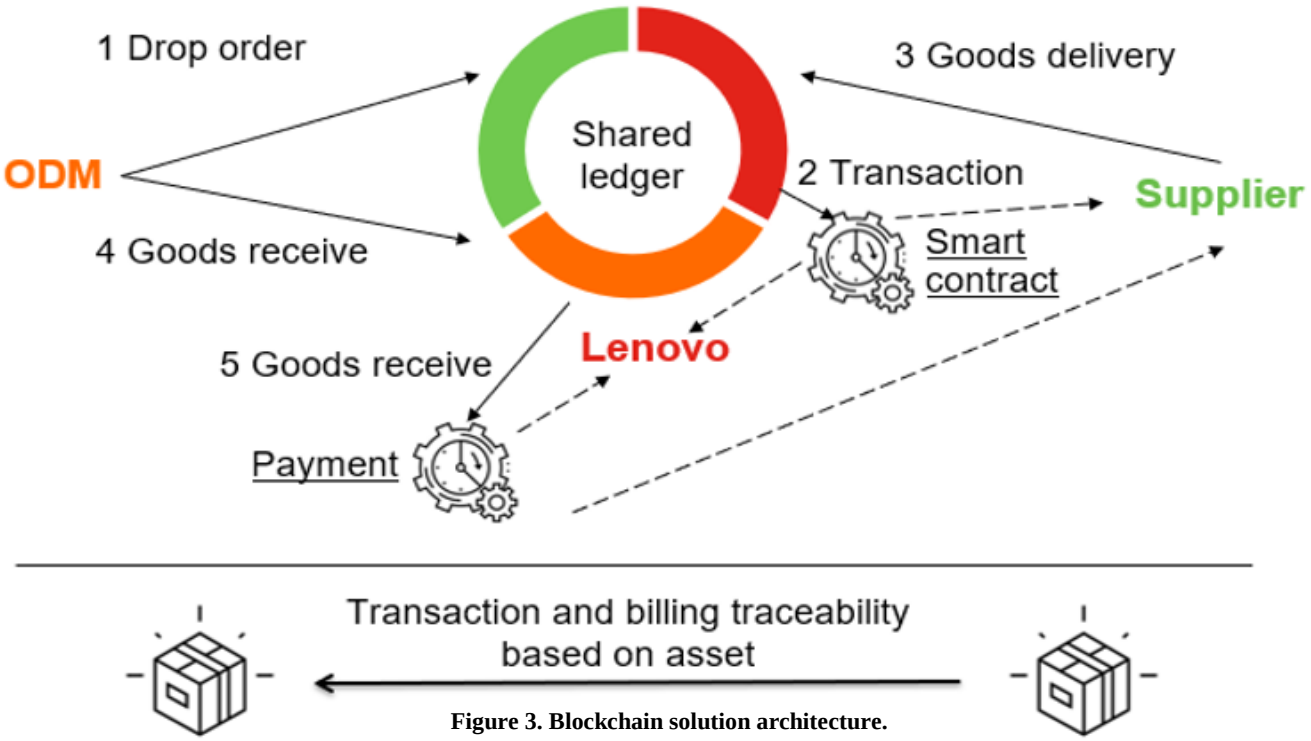


Figure 3. Blockchain solution architecture.

This blockchain based solution improves the whole business process data transparency for all participants. Supplier become aware of the ODM's requirements submitted to the chain and begin to stock-up as soon as possible. ODM can also determine Lenovo and supplier's order processing status at any time. One consensus ledger helps to increase data consistency. As a result, Lenovo reduces human input in data verification and communication. Blockchain is a decentralization system, in which consensus algorithms such as RAFT and BFT are involved to solve a subset of nodes crash fault and non-crash fault. Each participant's operational efficiency has been improved. The collaboration efficiency of this supply chain network also has been improved.

Our experience with implementing this solution has resulted in several lessons learned. The need to get all participants onboard with the solution is critical. Each of the participants should understand the benefits of joining the private, permissioned business network so that they can justify the investment in implementing a blockchain based solution. Furthermore, selecting a technology platform that is mature enough to allow a quick proof-of-concept (PoC) implementation and also provide mechanisms for incorporating extensions easily was a key requirement. For these reasons, the Lenovo Blockchain Platform was developed based on the Hyperledger Fabric with some enhancements to address the requirements for data privacy.

We are currently undertaking additional assessment of the solution along both business and technology dimensions.

From a technology perspective, ease of implementing smart contracts and associated business logic, ability to add specific security and audit related capabilities, and leveraging the data model for incorporating some machine learning techniques are the main evaluation criteria. From a business perspective, we are assessing the cost of the implementation, the improvements in operational efficiencies and the associated reduction in costs as well as the non-tangible gains resulting from a faster, more trusted business process.

5. CONCLUSION

Although there is a lot of research on supply chain management and supply chain integration, enterprises are still facing a significant number of challenges in supply chain information exchange because of the increasing size and complexity of their supply chain networks. In our case, we use blockchain technology to generate a supply chain alliance and provide many-to-many connectivity to solve the issue of point-to-point and one-way transmission of information. The shared ledger helps to improve data transparency. Consensus and immutability of blockchain ensures data consistency and credibility in supply chain.

Next, we will continue to study supply chain business expansion on blockchain, involving carrier and customs department in the alliance to improve logistics visualization and fast track, and financial institution to develop supply chain finance business, involve government for E-billing realization (see Figure 4).

Figure 4. Blockchain solution expansion.

The readers can use this solution to build their supply chain alliance, improve data transparency, collaborate with partners and transform to a data-driven business model, use blockchain technology to build a win-win business ecosystem. The reader can also follow our methodology for blockchain project implementation, start with simple scenario, a small number of partners, consider the common benefit of the alliance not only for the core enterprise. After the small alliance runs well, then expand the business scenarios and involve more partners to build the industry alliance.

At the same time, blockchain technology itself is evolving. There are some limitations in the large-scale application of supply chain, such as in some high frequency trading scenarios where the blockchain performance in terms of TPS (Transaction Per Second) needs to be improved. Cross-chain technology needs further development. In the future, the blockchain technology combined with IoT, AI and other technologies will realize more comprehensive supply chain optimization.

6. REFERENCES

[1] Nakamoto S. 2008. Bitcoin: A peer-to-peer electronic cash system. Consulted.

[2] D.G FENG, Y.G OUYANG. 2018. Preface of Special Issue on Blockchain Technology. Journal of Cryptologic Research, 5(5): 455-457.

[3] Xiaheng Zhang. 2018. Optimization of supply chain management mode based on blockchain. China's Circulation Economy, (8): 42-50.

[4] Zheng He. 2015. Web services-based cluster supply chain information integration. Laboratory research and exploration,34 (1): 107-112.

[5] Jie Zhou, Wenyu Li, Gang Guo. 2017. Analysis of Patent Situation of Blockchain Technology. Telecommunication Network Technology, (3): 37-42.

[6] Chuanlei Wang, Yiwei Wan, Qin Qin, Ningning Wang. 2017. A Resource Chain Logistics Information Ecosystem Model Based on Blockchain. Information Theory and Practice, (7):115-121.

[7] CHAVEZ- DREYFUSS, G.IBM. 2017. Maersk in blockchain tieup for shipping industry. Retrieved from http://www.reuters.com/article/us-usa-Blockchain-ibm-idUSKBN16D26Q

[8] Langley, C., Coyle, J., Gibson, B., Novack, R., & Bardi, E. 2008. Managing Supply Chains: A Logistics Approach. Canada: South-Western Cengage Learning.

[9] Janvier-James, A. M. 2012. A New Introduction to Supply Chains and Supply Chain Management: Definitions and Theories Perspective. International Business Research. 5,1 (Jan. 2012), 194.

[10] Awaysheh, A., & Klassen, R. D. 2010. The impact of supply chain structure on the use of supplier socially responsible practices. International Journal of Operations & Production Management, 30(12), 1246-1268.

[11] Christie Schneider. 2016. The biggest data challenges that you might not even know you have. Retrieved from https://www.ibm.com/blogs/watson/2016/05/biggest-data-challenges-might-not-even-know/

[12] Jeanette Barlow. 2018. How AI is Transforming the Supply Chain. Retrieved from https://www.ibm.com/blogs/watson-customer-engagement/2018/09/25/how-ai-is-transforming-the-supply-chain/

[13] IBM Supply Chain Insights. Retrieved from https://resources.lightwellinc.com/hubfs/downloads/IBM-Solution-Briefs/sb-IBM-Supply-Chain-Insights-(SCI)-Solution-Brief.pdf

[14] National Research Council. 2000. Surviving supply chain integration: strategies for small manufacturers. The National Academy of Sciences, p. 27.

[15] Yuzhu Liu, Chen Li, 2013. Study on the integrated framework of supply chain core enterprise data center based on soa. Information Management, Innovation Management, and Industrial Engineering.

[16] Patil, M., 2015. Challenges for Supply Chain Management in Today's Global Competitive Environment. constraints, 7(10).

[17] Lei Wen, Xiaojuan Zhang. 2010. Intuitionistic Fuzzy Group Decision Making for Supply Chain Information Collaboration Partner Selection. International Conference of Information Science and Management Engineering.

[18] Wüst, K. and Gervais, A. 2018. Do you need a Blockchain?. In 2018 Crypto Valley Conference on Blockchain Technology (CVCBT) (pp. 45-54).IEEE.

Elastic Smart Contracts across Multiple Blockchains

Schahram Dustdar
Distributed Systems Group
TU Wien
Austria
dustdar@dsg.tuwien.ac.at

Pablo Fernandez
Applied Software Engineering Group
Universidad de Sevilla
Spain
pablofm@us.es

José María García
Applied Software Engineering Group
Universidad de Sevilla
Spain
josemgarcia@us.es

Antonio Ruiz-Cortés
Applied Software Engineering Group
Universidad de Sevilla
Spain
aruiz@us.es

[Vision short paper]

ABSTRACT

In this paper, we deal with questions related to blockchains in complex Internet of Things (IoT)-based ecosystems. Such ecosystems are typically composed of IoT devices, edge devices, cloud computing software services, as well as people, who are decision makers in scenarios such as smart cities. Many decisions related to analytics can be based on data coming from IoT sensors, software services, and people. However, they typically are based on different levels of abstraction and granularity. This poses a number of challenges when multiple blockchains are used together with smart contracts. This paper proposes to apply our concept of elasticity to smart contracts and thereby enabling analytics in and between multiple blockchains in the context of IoT.

Categories and Subject Descriptors

• **Computer systems organization~Architectures**

Keywords

Elastic Smart Contracts; Internet of Things; Blockchain; Virtual Chains, Smart Cities.

1. INTRODUCTION

Cities are complex ecosystems, and their effective and efficient functioning has enormous impact on the quality of life of their citizens and society as a whole. However, building smart cities is probably one of the most difficult challenges our society faces today. Among the variety of problems that need to be solved, the question of how to leverage existing ICT technologies to develop foundations for smart city analytics in a transparent and trustworthy form greatly concerns all stakeholders in today's smart cities.

Second International Symposium on Foundations and Applications of Blockchain (FAB '19)
April 5, 2019, Los Angeles, California, USA.

As of today, we have observed several technologies enabling the connection between social and technical subsystems for smarter city analytics. A huge number of Internet of Things (IoT) devices as well as human participation have been introduced to provide various types of data about urban mobility and transportation systems, electricity grid, smart buildings, manufacturing, intelligent logistics systems, and critical infrastructures. Cloud systems have been introduced and used to store and analyze these big "volume, variety, velocity and veracity" streaming things-based and social data through complex middleware for various analytics needed for the operation and optimization of cities. Human capabilities have been invoked in the loop to design and monitor cities together with software. All of these data, analytics capabilities, and domain knowledge in smart cities are involved by a large number of stakeholders, ranging from individual citizens, corporates, to government agencies for both vertical and horizontal problems (such as energy consumption analytics or human mobility analytics). In this view, one needs to understand that analytics of smart cities are far from just "big data analytics" and IoT data analytics. Smart cities analytics have an inherent ecosystem requirement, leading to different paradigm shifts in big data analytics from transactions to ecosystem perspectives as well as in the involvement of multiple, not necessarily trusted stakeholders besides ICT sensors, networks and analytics.

Key city analytics often require data, analytics, and capabilities from both vertical and logical domains (e.g., related to energy consumption) in a complex ecosystem of things, software services, and people with multiple stakeholders, with varying trustworthiness degrees. Complexities in these analytics can be viewed by multiple stakeholders from different angles: **(i)** ***physical (space) view*****:** city analytics can be carried out for a single block, a street, or a house, **(ii)** ***logical domain view***: city analytics are needed for various vertical domains (e.g., building management, intelligent transportation management, and infrastructure maintenance) and horizontal domains (e.g., energy policy and governance, social wellbeing, and urban planning), and **(iii)** ***time view*****:** city analytics can be performed at different time-scales, e.g., online (with near real-time streaming data), offline (with historical data), as well as a combination of both near real-time and historical

data, also considering accountability aspects. While current data gathering techniques are able to collect various types of data, state-of-the-art analytics techniques isolate data produced by technical systems (e.g., from sensors) and social systems (e.g., from people) and then centralize the data in centers (e.g., in clouds) to carry out analytics at centralized places (although utilizing parallel and distributed computing resources). Such approaches rely entirely on software capabilities to deal with big data captured through distributed hierarchical networks of computing elements. In city analytics, data, information, knowledge, and computational capabilities from software services, things, and people are distributed in deep, interwoven distributed ICT architectures. Therefore, state-of-the-art approaches are not adequate as they collect data at the edge of the city where things and people reside, bring the data to the root of the hierarchy (e.g. cloud), and perform analytics based on data provided by predefined settings. First, it does not support time-scale because fine-scale and coarse-scale data analytics are not interoperable, as either we miss a lot of data (in coarse-scale data) or we have to deal with lots of data (in fine-scale data). Second, this also makes the filtering and pre-processing data challenging for supporting complex logical domains, which must deal with different logical horizontal and vertical scales. Finally, we also have severe problems with physical scale: as most of the time we centralize data in one cloud data center so we don't have enough information to cover all physical spaces with sufficient quality to guarantee time-aware analytics, e.g., subjects to be analyzed change rapidly in physical world and we lack up-to-date information in the centralized computing environment.

We believe we need *flexible and elastic mechanisms to support city analytics by harnessing collective capabilities of things, people, and software to carry out timely, quality-aware, and elastic analytics spanning both horizontal and vertical domains*. Given the huge number of things, people, and software services easily to be found and utilized without the need of centralized control, we should investigate a fundamental paradigm shift in utilizing collective capabilities that are distributed across the city infrastructure to enable coordinated analytics in a flexible and elastic manner. Such analytics must be provided with adjustable quality of results for multiple stakeholders where complex, transparent, and trusted collaboration between things, software, and people is needed to understand and address past, current, and future problems of smart cities based on historical, current, and predicted data.

In this paper, we discuss to what extent blockchain technologies are adequate to support complex analytics in these ecosystems. We first introduce a concrete motivating scenario in smart cities analytics (Sec. 2) and analyze how existing approaches to smart contracts and virtual chains can be applied to carry out the relevant analytics (Sec. 3). Our vision, described in Sec. 4, further develops the smart contract notion towards an *elastic smart contract*, which considers elasticity concerns, while providing a framework to horizontally and vertically integrate data and its associated analytics capabilities by promoting the idea of *glue contracts*. In Sec. 5 we conclude that our envisioned proposal will provide a comprehensive support for the different capabilities required in complex scenarios like smart cities.

2. MOTIVATING SCENARIO

As depicted in Figure 1, in this paper we consider smart city infrastructures consisting of (a) IoT sensors, (b) edge devices, which perform computational tasks such as analytics tasks, (c) more "powerful" edge servers (aka fog computing nodes), and (d) cloud computing data centers as the fundamental architectural building blocks for sensing and processing IoT data in smart cities.

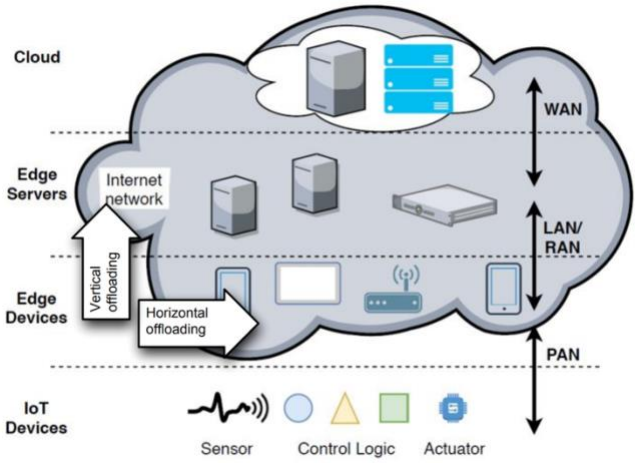


Figure 1. Vertical offloading of analytics computation [1].

At the lowest level of the current smart city infrastructure, we see that data flows from the edges to the data center. From the infrastructure perspective, at the edge (e.g., buildings or districts) we can identify numerous capabilities offered by things, software services, and even people. At (and through) the data centers, several types of software services and people (from the crowds, professional groups, etc.) are available to perform data management and analysis. Although various types of infrastructures connecting people, IoT, and software services are distributed, current city analytics processes are mainly performed in the cloud using software services to provide results to humans. In principle, analytics processes can be carried out in multiple places within the city infrastructure by leveraging the collective capabilities of units of IoT, people, and software services. However, with today's techniques, such units cannot be collectively composed and provisioned on the fly for subsequent distribution throughout the city infrastructure. This prevents us from providing timely and elastic analytics to support non-functional concerns, such as cost, security, and privacy.

For complex problems, city analytics processes are logically divided into a set of sub-analytics processes that cover a set of concerns in distinct horizontal and vertical domains, as shown in Figure 1. Computational tasks can be structured in a "vertical" way or in a "horizontal" way. Given the exemplified city analytics process for policy and regulation of sustainable environments, let us consider an analysis for a city block. Sub-analytics process concerns could be energy consumption of buildings and infrastructure, citizen wellbeing and opinions, environmental impacts of regulations, or incentive policies for green businesses, to name just a few. These sub-analytics processes belong to different vertical and horizontal domains and we need to correlate them and their results in order to understand how to create policy and how to regulate sustainable environments. In principle, such sub-analytics processes are also complex and some of them will be carried out in the cloud (such as, environmental impacts, and incentive policies) whereas others can be performed at the edge where things and people reside (e.g., building energy consumption, and citizen wellbeing and opinions). They also require different algorithms, data, and knowledge from different stakeholders. Among them, there are different ways to exchange analytics results and requests to ensure the final result of the city analytics to be delivered. To the best of our knowledge, state-of-the-art techniques just focus on centralized analytics for single domains. This leads to a severe problem for city analytics: as the scope of current analytics processes is limited to isolated domains and problems are either solved by software services or people, the results may not be adequate and substantial in the overall context of a city. We argue that smart city analytics must be researched from the perspective of ecosystems in which capabilities to contribute to analytics processes are based on hybrid resource types composed of software, people, and things. Moreover, different stakeholders from multiple vertical and horizontal domains impose requirements on analytics processes due to the associated ecosystem of people, technology, and institutions.

Analytics processes in smart city applications can therefore be performed along two dimensions: horizontally, e.g., monitoring and controlling across a number of different domains (and edge or IoT infrastructures) and vertically, e.g., performing analytics processes for particular domains such as buildings in a particular street, as depicted in Figure 2.

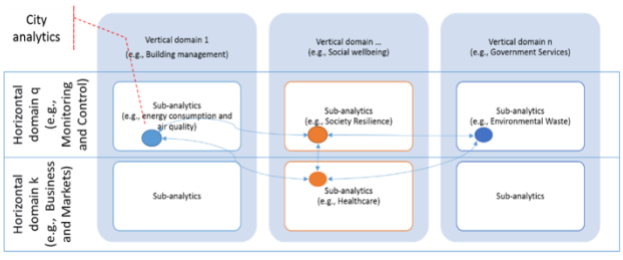


Figure 2. City analytics – logical horizontal and vertical sub-analytics, domains, and stakeholders.

In this scenario it is crucial that the shared data used to perform decentralized analytics in any dimension comes from trusted sources. However, considering the number of agents and stakeholders participating in a smart city ecosystem, trustworthiness cannot be assumed. Furthermore, certain stakeholders, such as public administrations, usually require transparency and tamper resistance to the data they use to analyze and provide services to other agents. For instance, a local administration may enact a contract with an external company to provide street cleaning services, using data from IoT sensors and possibly edge devices located on the streets to plan the optimal cleaning routes. Both the input data and the cleaning routes derived from its analysis should be publicly accessible in a transparent and immutable form, so that the local administration or even citizens can check whether the street cleaning company adheres to the contract in place and the quality level of the provided service, while providing flexibility and adaptability to changes in the ecosystem.

3. RELATED WORK

3.1 Smart Contracts

In a complex scenario like the introduced before, a variety of stakeholders have to collaborate, sharing information between them and allowing each party to carry out analysis and provide decentralized services over the shared data. Trust issues become fundamental in this setting, since parties have to continuously agree on the validity of the data and services they need to integrate. Blockchain technologies are a natural fit, providing transparency and non-tampering to the data shared in a trustless network [2]. In addition to these features, privacy and rights management can be considered by using different blockchain implementations, ranging from permissioned blockchains [3] to specific solutions tailored to IoT-based ecosystems [4].

Since the introduction of smart contracts [5], blockchains have evolved from mere distributed digital ledgers to distributed computing platforms that can include not only an immutable data repository, but also logical and behavioral information to automatically rule the relationships between stakeholders. Thus, smart contracts can encode functionality needed to provide additional services on top of the data registered in the blockchain. These contracts essentially aggregate some data under certain conditions that will trigger its execution. Although the data used within the contract logic is mostly obtained from the blockchain where the contract is deployed, oftentimes there is a need to consider external data (commonly referred as off-chain data). In order to retain the trustless characteristic of blockchains, an additional agent, namely an oracle, needs to provide the external data in a secured, trusted form [6].

3.2 Virtual Chains

Furthermore, there are scenarios where there is a need to separate nodes and information between different levels, as in our motivating scenario (see Sec. 2). Virtual blockchains provide means to implement specific functionality on top of existing blockchains [7]. They introduce an abstraction layer on top of existing blockchains, so that the different application nodes subscribing to the virtual chain will access data and execute smart contracts tailored to their characteristics, while using a single blockchain as the backbone for recording every transaction within the whole system. Thus, multiple virtual blockchains (or virtual chains for short) comprising the different levels discussed in our motivating scenario can be deployed and integrated using this approach. However, sharing data between different virtual chains and from off-chain sources still needs the introduction of oracles, which could be just rights management systems in case of internal oracles allowing data access between virtual chains deployed on the same regular blockchain.

3.3 Elasticity

As the complexity of the systems grows, the need to adapt to variable flows of information and constraints to develop appropriate outcomes represents an important challenge. To this concern, elasticity is presented as the capabilities to react and accommodate changes in the environment with an autonomous mechanism. In [8], authors provide a formal model of elasticity as a three-dimensional space involving resources, quality, and cost aspects that provide the appropriate framework to define and analyze the elasticity properties of an information system that will be used as a starting point of our conceptual proposal.

4. CONCEPTUAL PROPOSAL

Smart contracts represent an appropriate framework to develop a computational mechanism combining data off-chain with the one present in the blockchain. However, in order to address the analytical challenges discussed in the motivational scenario, the framework should be extended to support a variable and multilevel nature of the actors involved. Specifically, in this section, we outline how the elasticity and integration aspects are fundamental cornerstones to build an appropriate smart contract ecosystem to develop more capable blockchains for complex scenarios such as a smart city.

4.1 Integration Concerns

Separating the information needs in different levels allows organizations to focus on their interests, while regulatory bodies can grant access to those organizations only to specific data. In this context, from a blockchain perspective there are multiple architectural alternatives to implement the level stratification, which can be characterized by analyzing three aspects:

- **Granularity**. Several mapping options could be defined to assign a given blockchain to a single level (fine granularity) or multiple levels (coarse granularity). In addition, there could be some scenarios where the same levels are composed of multiple different blockchains
- **Accessibility**. From this perspective, we refer to the capability to analyze the blockchain content by different agents; i.e., the blockchain represents an open system (public) to any agent or a closed system (private or permissioned) to certain agents.
- **Deployment Model.** In this context, we address the logical implementation and deployment of the chain: existing blockchains that are implemented over a specific technical protocol or virtual chains that are materialized inside a regular existing blockchain.

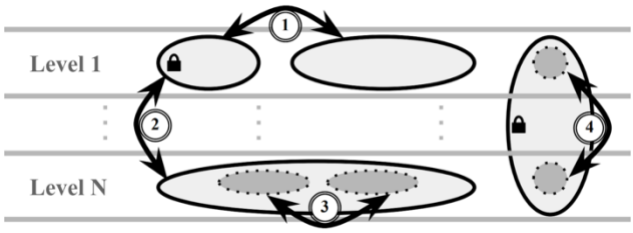


Figure 3. Integration points between blockchains.

Consequently, we can have a wide variety of modelling choices for a given scenario; exemplary, Figure 3 depicts a particular abstract scenario showing several options: level 1 with two fine grain blockchains (one private and one public), level N with one fine grain public blockchain that contains two virtual chains, and a coarse grain private blockchain that spans over all levels and contains a virtual chain for each level. From an analytics perspective, since smart contracts are meant to be executed in the context of a single blockchain, we envision the need for different cross-chain integration mechanisms (as exemplified in Figure 3) depending on three factors: whether integration is done between regular blockchains (Examples labeled with 1 and 2) or virtual chains (3 and 4); between chains in the same level (1 and 3) or different level (2 and 4); or between the same accessibility context (3 and 4) or between a public and a private chain (1 and 2). Taking these challenges into account, we claim the need for a special kind of smart contracts, coined as *glue contracts*, with the special responsibility of making data available across two different chains (virtual or regular) corresponding to the same level (horizontal integration) or different levels (vertical integrations).

In this context, it is important to highlight that integration options presented would represent different types of glue contracts: as an example, in order to integrate two different chains, a possible solution could make use of oracles in order

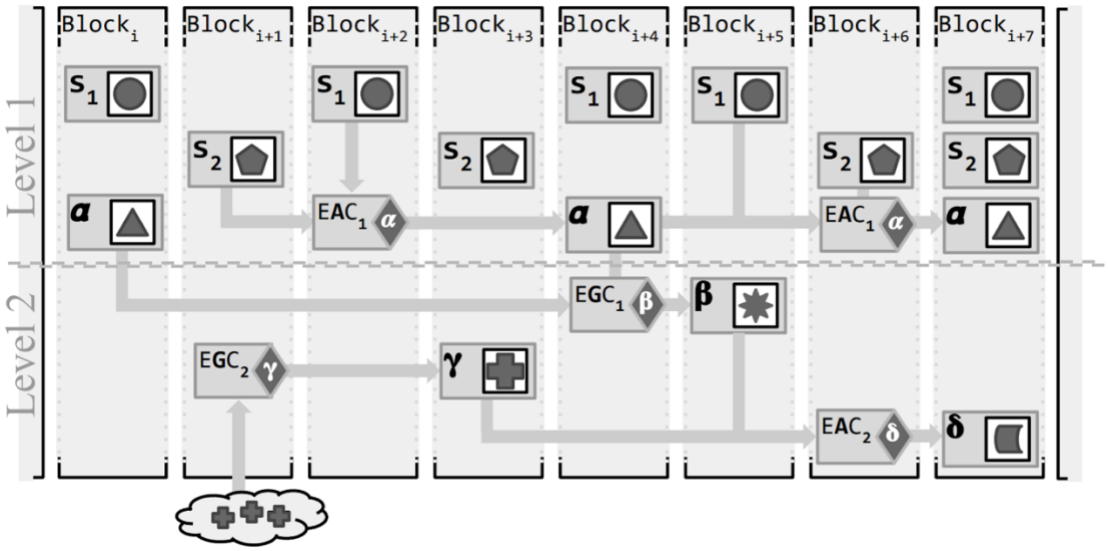


Figure 4. Blockchain fragment example for the use case.

to maintain the trust level of the whole ecosystem; in this particular case, the software oracles are just simple gateways to the accessed blockchains which do not need to add an additional trust method to the already trusted data from the accessed blockchain. Another example of mechanism used by glue contracts to address an integration between accessibility contexts could be the usage of IPFS[1] as the intermediary persistence area for data. In the case of cross-level (or vertical) integrations, glue contracts would be in charge of aggregating the data from inferior levels into new kind of information for higher levels. Furthermore, glue contracts need to address possible divergences between blockchain implementations and protocols of chains to be integrated. There exist alternatives to reconcile these divergences when dealing with crypto currencies [10, 11] that could be extended to allow dealing with complex asset integration.

4.2 Elasticity Concerns

Following the model presented in [8], the envisioned proposal takes into account the elasticity concerns to allow stakeholders to dynamically reconfigure the integration between levels, depending on the horizontal and/or vertical offloading needs (i.e. contract execution), by leveraging elasticity for analytical and glue contracts, correspondingly.

Specifically, in order to incorporate the elasticity dimensions in smart contracts, we need to provide means to elastically define resources, quality properties, and costs associated with a particular contract. To this end, we propose to add an abstraction layer to current smart contracts which will define the elasticity policies for a particular contract. Therefore, executing a so called *elastic smart contract* will transparently consider elasticity aspects on top of the actual functionality provided by the contract. Furthermore, stakeholders should consider executions costs for contracts (e.g. gas for Ethereum smart contracts) as well as infrastructural costs of the blockchains to plan the actual architecture of chains in levels; to this end, a decentralized market of agents [9] would allow the dynamic reconfiguration of the ecosystem taking cost information into account.

4.3 Visionary Use Case

To exemplify the applicability of the proposal we outline a supporting architecture grounded on the current capabilities of blockchain technological state of the art. In such a context, in the current evolution state of the technology towards richer ecosystems, we expect continuous improvements and revisions of the conceptual frameworks presented. In this use case, (Figure 4 shows a fragment of the envisioned blockchain) we can conceptualize an architecture of different virtual chains (composed of "virtual" blocks) that coexist in the same blockchain ecosystem (composed of "grounded" blocks) with smart contract capabilities (such as Ethereum). In such a framework, each grounded block would be a container for multiple virtual blocks that correspond to the different levels and contain either data or contracts related to that level. Specifically, in Figure 4 we exemplify a fragment of the blockchain (Blocks i to i+7)) including two levels (note that in a real scenario there potentially exist a higher number of levels): inside Level 1 we can identify information generated by two agents (s_1 and s_2) and one elastic analytical smart contract (EAC_1) in charge of creating derived data from the activity in the level. Next, in Level 2, we can see two kinds of elastic glue contract

[1] https://ipfs.io/

(EGC): on the one hand, EGC_1 aggregates the information from Level 1 and incorporates the aggregation as new data in Level 2; on the other hand, EGC_2 (implemented as an oracle) imports data off-chain to the Level 2. Finally, we can see EAC_2 analyze the data of the level to create a new kind of information.

In this context, we can identify different examples of multiple interleaved analytics that can be mapped to the abstract blockchain fragment presented in Figure 4: from low-level analytics regulating small physical spaces that mainly involve sensor data to high-level analytics involving other kind of data sources such as human actor decisions or off-chain census data. For the sake of clarity, we propose a simple example that would correspond with two low levels of analytics representing an adaptable urban lighting system:

Level 1 (street section) would represent a section of a street composed by a number of sensors and lights; concretely in the chain fragment depicted, agents s1, s2 could represent two presence sensors for a given road section that introduce their observations as data in the chain with different time resolution. The analytics contract EAC_1 would periodically perform an analysis over the sensors data to calculate a presence prediction ($\boldsymbol{\alpha}$) in the section; this analytical information would be used to actuate into adaptable street lights in the street section that switch on in the presence of cars, so they dynamically adapt their switch-off latency to the actual prediction.

Level 2 (street) the glue contract EGC_1 could aggregate the presence prediction of different sections calculated in Level 1 in order to create an estimation of the traffic flow in the street ($\boldsymbol{\beta}$); in this level the glue contract EGC_2 could include weather forecast as off-chain data ($\boldsymbol{\gamma}$) so the analytics contract EAC_2 could calculate an estimation of the congestion risk ($\boldsymbol{\delta}$) in order to optimize the traffic lights rules for the given the street.

Furthermore, in a potential superior *Level N* we could leverage advanced use cases such as a new generation contract for waste management service that regulates the actual resource assignment algorithm based on the data harvested by the sensors; this could be implemented by a combination of elastic smart contracts using the analytics gathered and calculating the actual bills automatically having a total transparency and non-tamper management procedure.

Examples of the three elasticity dimensions emerge from our use case: (i) *resources* range from the information providers that can correspond with things (e.g., sensors), software (e.g., government information systems) or people (e.g., an approval from a stakeholder); (ii) depending on the type of resource, a taxonomy of *quality* aspects can be defined (such as resolution data in sensors, availability of the government information system or readiness of the stakeholder); (iii) finally, *costs* involved in the process can also be structured in terms of the resource type (e.g. energy cost of the sensor, infrastructure cost of the information system, or personnel costs). All these concerns would be taken into account to create the elasticity policies for each elastic contract; as an example in the use case, EAC_1 would have a policy to select the number of sensors (resources) filtered by a particular data frequency (quality) and constrained by a maximum number of gas used in the execution of the analytics (cost).

5. CONCLUSIONS

When facing complex scenarios as those that arise in smart cities, where transparency and accountability of the data and analytics are key goals, blockchains are a natural fit. However as these scenarios are typically composed by a complex ecosystem of IoT sensors, edge devices, fog nodes, and cloud data centers, the application of traditional blockchain technologies poses several challenges concerning elasticity and integration aspects, since the requirements for the analytics to be performed varies dynamically, not only in terms of resources needed, quality and cost aspects, but also in the dimensions of those resources. Thus, in order to support elasticity as well as horizontal and vertical integration, in this paper we introduce the concept of elastic and glue smart contracts.

The evolution of current blockchains towards supporting our envisioned elastic smart contracts needs the introduction of elasticity related information to the contracts logic. We propose an elasticity policy abstraction layer to extend the existing smart contracts introducing rules to account for variations in the three elasticity aspects (resources, quality and cost). Additionally, we characterize the different integration scenarios that can be applied to elastic smart contracts, exemplifying them in the context of smart cities. Our vision is that using approaches such as virtual chains and adapting current elastic services frameworks, we can achieve a greater level of integration inside (and between) the various analytical levels while keeping a flexible reconfiguration of the architecture in case there is a need for vertical or horizontal offloading of computation.

6. ACKNOWLEDGMENTS

This work has been partially supported by the EU Commission (FEDER), Spanish and Andalusian R&D&I programmes under grants TIN2015-70560-R and P12-TIC-1867, and the TU Wien Research Cluster Smart CT.

7. REFERENCES

[1] Gusev, M., Kotoska, M., Jakimovski, B., Dustdar, S., Scekic, O., Rausch, T., Nastic, S., Ristov, S., Fahringer, T. (2018). An Architecture for Streaming IoT Applications: A Deviceless Edge Computing Approach. under submission

[2] Swan, M. 2015. *Blockchain: Blueprint for a New Economy*. O'Reilly Media.

[3] Androulaki, E. et al. 2018. Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. *Proceedings of the Thirteenth EuroSys Conference on - EuroSys '18* (New York, New York, USA, 2018), 1–15.

[4] Dorri, A., Kanhere, S.S. and Jurdak, R. 2017. Towards an Optimized BlockChain for IoT. *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation - IoTDI '17* (New York, New York, USA, 2017), 173–178.

[5] Ethereum Foundation, 2018. *A Next-Generation Smart Contract and Decentralized Application Platform*. [Online, last accessed: 7-Dec-2018]. Available: https://github.com/ethereum/wiki/wiki/White-Paper

[6] Xu, X., Pautasso, C., Zhu, L., Gramoli, V., Ponomarev, A., Tran, A.B. and Chen, S. 2016. The Blockchain as a Software Connector. *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)* (Apr. 2016), 182–191.

[7] Nelson, J., Ali, M., Shea, R., and Freedman, M. J. 2016. Extending existing blockchains with virtualchain. *Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL'16)*, (Chicago, IL).

[8] Copil G., Moldovan D., Truong H.-L., Dustdar S. (2016). rSYBL: A Framework for Specifying and Controlling Cloud Services Elasticity. *Transactions on Internet Technology*, Volume 16, Issue 3, pp. 18:1 - 18:20

[9] García, J.M., Fernandez, P., Ruiz-Cortés, A., Dustdar, S. and Toro, M. 2017. Edge and Cloud Pricing for the Sharing Economy. *IEEE Internet Computing*. 21, 2 (2017), 78–84.

[10] M. Herlihy. 2018. Atomic cross-chain swaps. arXiv:1801.09515

[11] Hope-Bailie, A. and Thomas, S. 2016. Interledger: Creating a Standard for Payments. *Proceedings of the 25th International Conference Companion on World Wide Web (WWW '16 Companion)*, 281-282.

Smart Contracts and Demurrage in Ocean Transportation

Haiying Jia
SNF Center for Applied Research, 5045 Bergen, Norway
and
Center for Transportation and Logistics, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA
+1 617 401 5231; Haiying.jia@snf.no

Roar Adland
Department of Business and Management Science, Norwegian School of Economics, 5045 Bergen, Norway
and
Center for Transportation and Logistics, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA
+1 617 2587 311 roar.adland@nhh.no

ABSTRACT
We explore how blockchain-based smart contracts may automate the monitoring and execution of demurrage clauses in logistics. Building on the legal framework for the ocean transportation of bulk commodities, we outline the benefits and challenges in streamlining the demurrage process. Our findings suggest that while many of the contractual clauses relating to demurrage can be resolved algorithmically by remote sensing data, the need for subjective human opinion remains. The main challenge in adopting smart contracts is the reliance on 'trustworthy' off-chain resources and the difficulties in aligning the interests of participants in the system. Our analysis is important as an input to ongoing industry initiatives in the design of blockchain applications for supply chain management.

Categories and Subject Descriptors
Application use cases

Keywords
Smart contracts, blockchain, logistics, demurrage, charterparty

1. INTRODUCTION
Transportation contracts typically include a "laytime and demurrage" clause in order to allocate the cost of delays caused by prevalent risks such as terminal congestion or strikes, in addition to the typical case of cargo being delayed. The term demurrage, which originated in ocean transportation and now extends to other transportation modes, refers to the "penalty payment" for the extended time period that the transportation capacity (be it a vessel, container or railroad car) remains in possession of the charterer (shipper) after the agreed period allowed for loading and unloading (laytime). Accordingly, demurrage is a potential payment from the user of the transportation asset to its disponent owner. It is a source of revenue used to offset per diem on transportation capacity held solely for the benefit of customers and thus can be viewed as extended freight (Jia and Adland, 2018).

The occurrence and realization of demurrage is subject to conditions and provisions that are outlined in the contract. Complicating factors include operational procedures, such as when to give Notice of Readiness (NOR) to commence laytime (i.e. the contracted time for loading and discharge), and the large number of stakeholders involved (shipowner, port authority, charterer, agents and/or cargo owner). Most importantly, contracts tend to use ambiguous language, creating disagreements over what is said in the laytime and demurrage clauses. Sometimes a comma can make a difference. For instance, a typical phrase may look like:

"Cargo to be furnished and received by ship at port of loading as fast as vessel can receive in ordinary working hours, and to be received from alongside ship at port of discharge as customary as fast as steamer can deliver in ordinary working hours."

As a consequence, demurrage is arguably one of the most disputed contractual terms in the transportation industry (Summerskill, 1989). In cases when disputes arise, the interpretation of these conditions is left to arbitrators, lawyers and the courts. This is not only a concern to the contractual parties directly involved in the transportation service, but also to importers and freight forwarders as it relates to documentation and clearance of goods at ports. For instance, there has been reports of increasing congestion in US seaports due to idle containers (Mongelluzzo, 2000a, b). Mongelluzzo and Bonney (2014) reported an increasing number of complaints by truckers and shippers about demurrage penalties in US ports. Indeed, there is currently a US shipper-driven petition seeking policy guidelines that would make it easier to challenge demurrage and detention (Bonney, 2018). Veenstra (2015) argues that demurrage (and detention) can cause a general delay in the global supply chain. As a consequence, it is recognized by industry organizations and individuals that improved clarity and precision is vital (Laffaye, 2013).

Second International Symposium on Foundations and Applications of Blockchain (FAB '19)
April 5, 2019, Los Angeles, California, USA.

Smart contracts can potentially resolve some of these challenges by virtue of reducing or eliminating ambiguities in the execution and encouraging better information sharing among stakeholders. A smart contract, the term of which was first coined by cryptographer Nick Szabo (Szabo, 1994), is a set of promises, specified in digital form, including protocols within which the parties perform on these promises (Szabo, 1996). It is a computer protocol based on if-then statements intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. When a pre-programmed condition is triggered, the smart contract automatically executes the corresponding contractual clause. Blockchain technology, with key features such as distributed consensus mechanisms and near-tamper-proof data records, provides an interesting platform for smart contracts, and may ultimately facilitate a move from automated contracts to truly autonomous smart contracts capable of self-execution and self-enforcement.

The objective and contribution of this paper is to explore the application of smart contracts to execute the laytime and demurrage clause in ocean transportation. We identify and discuss the various legal, technical, and business issues in relation to the use of blockchain-based smart contracts for managing laytime calculations and demurrage payments. In a wider context, our research also highlights the inefficiency caused by the concept of demurrage, both in terms of productivity, legal costs and environmental issues. We find that the main advantage of smart contracts is that they force the use of precise contractual terms in place of the current ambiguous common-law terms that are the source of most disputes. Their use may also lead to greater standardization of ocean freight contracts, reducing the time and cost for their negotiation and drafting. Our research is important for the design and evaluation of blockchain-based applications for the ocean transportation industry.

The remainder of the paper is organized as follows. Section 2 reviews legal framework for demurrage. Section 3 outlines the architecture for a smart contract on demurrage. Section 4 discusses the challenges and managerial implications. Section 5 concludes and presents challenges for future research.

2. THE LEGAL FRAMEWORK OF DEMURRAGE

A contract is an agreement between parties about rights and obligations (including prohibitions, such as exclusions for where a vessel may operate). These may be obligations for actions that named parties are supposed to take at various times, generally as a function of a set of conditions. Commercial contracts are meant to be mutually beneficial, so that one of the reasons for contracting is reallocating or sharing benefits and risks (Shavell, 2003). In ocean transportation, the contract for the hiring of a ship for either a certain period of time (a timecharter) or a voyage between pre-defined port pairs (a voyage charter) is termed a charterparty. The charterparty sets out the terms and conditions for the use of the vessel by the charterer (the buyer of the transportation service), For voyage charters, a key clause in the charterparty relates to laytime and demurrage. The 'laytime' defines the time period available for loading and discharge to the shipper (charterer) and is subject to terms used in the contract, while demurrage is the daily penalty payable by the charterer should laytime be exceeded. The crucial – and often contentious point – is therefore the conditions that need to be satisfied before laytime commences, pauses, and stops. Broadly speaking, commencement of laytime occurs when the ship has *reached the destination*, is *reachable* and ready for cargo operations (*readiness*), has tendered its Notice of Readiness (NOR) and the charterer has accepted such NOR.

In this section, we break down the legal terms and conditions associated with demurrage clauses and examine the aspects that are prone to disputes. We note that the topic of laytime and demurrage is so large that it merits a separate book on its own (see, for instance, Cooke *et.*, 2014), and so we here only touch upon the essential terms and conditions.

2.1 Destination

For the purpose of the demurrage calculation, the destination is the focal point where the allocation of responsibilities and risks occur. Therefore, the geographical boundary is of utmost importance and needs to be clearly defined. In water transportation, destination may refer to a port or a berth. In land transportation, it may refer to a station or a terminal. In reality, even when the individual contract specifies a destination, disputes still arise. For instance, in cases when a port is named as the destination in the charterparty, but there is congestion within the port area and the ship has to wait at other places in the port, does this count as reaching the destination so that laytime starts counting? In courts, such disputes can refer to the "Reid test", a crucial passage in the judgment of Lord Reid in *Oldendorff (E.L.) & Co. G.m.b.H. v Tradax Export S. A.* (1973)[1].

As an effort to reduce disputes in this regard, particularly in tramp shipping, BIMCO (2013) has published an updated set of terms on basis of the widely adopted Voylayrules 1993

[1] Lord Reid reads: "…. Before a ship can be said to have arrived at a port she must, if she cannot proceed immediately to a berth, have reached a position within the port where she is at the immediate and effective disposition of the charterer. If she is at a place where waiting ships usually lie, she will be in such as position unless in some extraordinary circumstances proof of which would lie in the charterer…… if the ship is waiting at some other place in the port then it will be for the owner to prove that she is as fully at the disposition of the charterer as she would have been if in the vicinity of the berth for loading or discharge."

(BIMCO et al. 1993) in conjunction with calculating the running of laytime and demurrage. BIMCO (2013) defines the area of port as a rather wide concept: "any area where vessels load or discharge cargo and shall include, but not be limited to, berths, wharves, anchorages, buoys and offshore facilities as well as places outside the legal, fiscal or administrative area where vessels are ordered to wait for their turn no matter the distance from that area."

The transportation facility, such as a ship or a container, needs not only to physically present itself at the destination, but also notification and acknowledgement needs to be sent and received by different parties. In ascertaining whether a ship has arrived at the port, the courts consider not only the views of users of the loading and discharging facilities but also the extent of the activities of the various port authorities (Summerskill, 1989). This also involves an acknowledgement that the extents of the legal, administrative, fiscal and geographical boundaries may be taken into account.

2.2 Reachable on arrival

To have arrived at the destination for the purpose of the laytime commencement, the transportation facility must not only be within the port, but also be *"reachable on arrival, "always accessible"* or *"at the immediate and effective disposition of the charterer"*, according to the terms typically used in transportation contracts (e.g. BIMCO *et al.* 1993, BIMCO 2013). These terms and conditions may cause different understanding among the contractual parties. For example, in *K/S Arnt J. Moerland v. Kuwait Petroleum Corporation of Kuwait* (1988), the ship arrived at the pilot station and gave her NOR. Owing to her high draught, the ship did not move to the commercial port area until four days later and a series of unexpected events followed resulting in demurrage. The degree to which a ship is "reachable" involves a great deal of interpretation and arbitration.

If a ship is not able to be *always accessible* or at *the immediate and effective disposition* of the charter, due to events such as bad weather, waiting for next tide, waiting for tug or pilot, congestion, restriction on night navigation, etc., the question becomes which party is liable for breach. The level of liabilities with regards to the degree of reachability is also dependent on the negotiation power and market conditions. Many standard charterparty terms place the risk of port congestion or delay in berthing on the charterer (Cooke *et al.* 2014).

2.3 Readiness

A ship, a container or a railroad car is ready to load or discharge, in the sense that carriers can give a proper NOR, when it is available for shippers to use. The vessel must be "ready in a business and mercantile sense" (*Armement Adolf Deppe v. John Robinson & Co.,* 1917). The requirement that the facility (a ship, a container or a railcar) is ready involves a distinction between mere routine formalities –those which do not prevent her being regarded as ready – and matters which will cause delay. The usual checklist for physical readiness for a ship include, but is not limited to, her having loading/discharging gear ready, and adequate supply of fuel and boiler water. It is common to provide "whether Customs cleared or not" and "whether in free pratique or not" (*AET v. Eagle Petroleum*, 2010).

Once the captain or shipowner tenders the NOR, the charterer or his agent needs to *officially* accept it in clear and unequivocal terms to the effect that the charterer treats the NOR as valid, irrespective of its actual status. Where a charterer or his agent "accepts" a NOR, which is in fact invalid, but his acceptance is unqualified, the charterer may thereafter lose the right to assert that invalidity (see, *Sofial v. Ove Skou Rederi,* 1976; *Surrey Shipping v. Compagnie Continentale,* 1978). It suggests that the acceptance of NOR is not simply an act of replying to an email; the charter is also recommended to actually check the *readiness* of the vessel.

2.4 The commencement of laytime

Once the seaworthy vessel has arrived at the designated destination and tendered her valid NOR which is officially accepted by the charterer, laytime starts counting. Typically, the charterparty specifies the time that laytime commences in words, for example: "laytime shall commence at 1 p.m. if NOR is given before noon, and at 6 a.m. next working day if notice given during office hours after noon." Shipowners sometimes may contend that laytime has begun even though there has not been compliance with the contract provision. For instance, if loading and discharging happens on Saturday (non-working day), is this Saturday included in the laytime? Court practice is somewhat ambiguous in this regard.

The agreed duration of laytime can be stated in many ways. Examples of usual terms include: agreed a fixed time, "running hours", weather working days, or working days of 24 consecutive hours. In some cases, laytime is defined by loading/discharging rates, such as "tons per hatch per day" or so many tons "per available or workable hatch per day". In the case of "weather working days" or "weather permitting", the law of nature plays an important role. It is generally understood that weather conditions include heat, cold, wind, fog and precipitation, and their immediate consequences such as waves or swell (Cooke et al. 2014). In the current framework, the contractual language is very descriptive, and some distinctions seem somewhat artificial. For instance, rain may not prevent or endanger the cargo operations as such, but presents a risk of damage to the cargo if the operation is continued.

2.5 Demurrage accounting

If the vessel is detained in loading or discharging beyond the agreed laytime (the *free time*) the charterer is in breach of charterparty and therefore the shipowner is eligible for demurrage payment, payable at a fixed rate per day (hour)

and pro rata. Surprisingly, the great majority of charterparties impose no express limit on the period of demurrage. This suggests that demurrage payment can be claimed, in theory, forever.

This no-cut-off period is also a source of disputes. For instance, in *MSC Mediterranean Shipping Company SA v. Cottonex Anstalt* (2016), 35 containers with cotton cargo remained uncollected for an extended period of time. The Carrier (MSC) brought a claim for over US$1 million in respect of container demurrage. The Court imposed a cut-off point in its decision, by saying "it would have been wholly unreasonable for the carrier to insist on further performance (demurrage payable)."

It may seem obvious that charterer is liable to pay demurrage. However, the ownership of the cargo can change at the loading port from the FOB (Free on Board) buyer, or at the discharging port from the CIF (Cost, Insurance and Freight) seller. It then seems to be the case that the bill of lading holders are liable for demurrage incurred at both loading and discharging ports. In Gencon 1976, the ambiguous "merchants" are liable for demurrage.

3. SMART CONTRACTS

Unlike conventional contracts that are established through written words, and enforced by actions, arbitration or courts, smart contracts are algorithms built as self-executing and self-enforcing computer programs (Szabo, 1994). While not a recent invention, advances in information technology – particularly the decentralized consensus architecture built around blockchains - has caused renewed interest in the concept. The term blockchain refers to a fully distributed ledger system for cryptographically capturing and storing a consistent, immutable, linear event log of transactions between networked actors (Risius and Spohrer, 2017). Built upon primary distributed ledger functionality, recent platforms such as Ethereum or Hyperledger comprise elements for managing a fully distributed network of peers, different cryptography-enabled consensus mechanisms for capturing and storing transactions, and programming languages to create smart contracts (Glaser, 2017). We note here that smart contracts need not be deployed on a blockchain but the shared features of the two suggest a good fit: Smart contract execution is triggered by a sequential occurrence of events involving nodes in an ecosystem, while a blockchain relies on a similar distributed system to generate a distributed, secure, sequential, immutable and consensus-based data structure. We note here that this structure is also aligned with the physical movement of a single ship or cargo in both time and space – it is by definition linear and sequential.

For the remainder of this paper we will discuss the application of smart contracts with the implied assumption that it runs on top of a distributed system of nodes where information can be sequentially and cryptographically stored and consensus on the business process can be reached in an automated fashion. We do not delve deeper into the technology discussion. However, we acknowledge that a key decision in the actual implementation would be whether to employ a private (permissioned) platform or open (public) blockchain solution. In a permissioned system only invited parties can participate, potentially limiting the scope to the stakeholders in any given contract, albeit with access to different functionality and authorization levels. This gives the participants greater control in terms of who can access data and reduces the well-known concerns relating to the scalability and energy consumption of large public blockchains. Open blockchains, on the other hand, encourage industry-wide standardization and adoption, reduce the duplication of development efforts, and decrease concerns relating to the dominance of a single player. Both already have real-world supply chain implementations (c.f. IBM/Maersk's TradeLens platform vs. CargoX for trade documentation), and both have their pros and cons. However, we do not take a stand in this important discussion here.

As seen in the previous section, the way in which traditional ocean transportation contracts are worded can often result in ambiguity. In many instances, ambiguous language (open terms) can make it easier for parties to enter in to a contractual arrangement, creating flexibility in terms of contractual performance (Gergen, 1992; Hadfield, 1984). The presence of some commercial flexibility can in fact be valuable in a physical system operating under great uncertainty, such as the global supply chain. However, ambiguity can also be used by parties to scuffle free from contractual conditions. Smart contracts can potentially provide a solution to this problem by incorporating provisions into computer code. In particular, we see two major potential advantages[2]: Firstly, while smart contracts may not reduce the need for interpretation of a complex physical situation in relation to the terms of a contract, the parties implicitly pre-agree on that interpretation by committing to execution of the contract by an associated set of smart contracts and associated external resources. This should reduce the time and cost in monitoring and enforcing the legal provisions of the laytime and demurrage clauses. Secondly, if the implementation of smart contracts lead to a de facto industry standard (of the contract and its interpretation) this will reduce the time and economic costs in negotiating and drafting legal provisions. These are the main economic arguments for adopting smart contracts in our context.

[2] We thank an anonymous referee for this interpretation.

3.1 Validity and enforcement

Despite differences in the civil law and common law system in the approach to contract formation, it is generally recognized the key elements in the formation of a contract include: (1) it is a mutual arrangement and (2) the agreement is enforceable by law (see, for instance, Shavel, 2003; Bag, 2018). Assuming that both the shipowner and charterer has entered willingly into the transportation agreement as a result of a standard search-offer-acceptance process, another requirement for the contract to be legally valid is that both parties mutually assent the contract, in the form of digital signatures[3]. In the case of smart contracts, such assent would be in the form of private and public encryption keys. We note here that a court may not regard a smart contract as either itself being a legal contract, nor having priority in specifying the contract over other paper-based representations of the agreement, or indeed over "reasonable" or precedent-based interpretations of agreements. However, there is precedent for courts to recognize enterprise software systems to perform and monitor contracts, which would be the main purpose of the smart contract in our use case. Digital signatures are still important for supporting those mechanisms.

Most countries now have laws governing digital signatures, for instance, the European Union's Electronic Identification, Authentication and trust Services (eIDAS) (EU, 2014), the Federal Electronic Signatures in Global and National Commerce Act ("ESIGN Act") and the Uniform Electronic Transactions Act ("UETA") in the US, the Electronic Signatures Regulation in the UK, and the Electronic Signature Law of the People's Republic of China. The United Nations has published the guideline under UNCITRAL Model Law on Electronic Signatures for countries to follow. Basically, the laws ensure that: if a law requires a signature, an electronic signature suffices; and if a law requires a record to be in writing, an electronic record suffices. Cryptographic signatures fit the definition of "electronic signature" contained in this category. Once a contract is concluded, i.e. offered and accepted electronically, it is legally binding and enforceable in a court of law (UETA, 1999). In a prescient acknowledgement of smart contracts, UETA even recognizes the validity of "electronic agents" - computer programs that are "capable within the parameters of its programming, of initiating, responding or interacting with other parties or their electronic agents once it has been activated by a party, without further attention of that party". Overall, it is not at all clear that a new legal framework is required to ensure the validity or enforceability of signatures, records, or contracts that use smart contracts. Instead, commentators worry that the types of legislation currently under consideration are not only unnecessary, but may serve to create confusion rather than clarity (Hansen et al 2018).

3.2 System architecture

While the early blockchain-based smart contract applications have been purely digital[4], their implementation in a logistics setting requires a very different interaction with the physical world. For instance, consensus on the existence and ownership of a Bitcoin is based on the "Proof of Work" protocol developed by Dwork et al (1993) and is done solely "on chain", i.e. without any external input other than the energy consumed for computing power. Clearly, verification and consensus on the geographical position and state of readiness of a ship can only be achieved with knowledge of the physical world. Consequently, smart contracts dealing with demurrage must be able to access the external ("off chain") data streams that are required to control their business logic.

This requirement introduces an important component into the smart contract ecosystem - the oracle. In computer science terms, an oracle is an interface that delivers data from an external source via a secure channel to the smart contract (Bashir, 2018). In the context of demurrage and laytime calculations, such external data will include satellite vessel location data, onboard vessel sensor data, the vessel's electronic logs, weather data and inspection reports. An oracle can also be another blockchain storing authenticated data. Importantly, the requirement to use oracles and "off chain" resources reintroduces the issue of trust and potential for providing inaccurate or manipulated data. We will revert to this discussion later.

In addition to oracles, the smart contract ecosystem incorporates the nodes of the blockchain itself, that is, a distributed network of computer servers that record the data (e.g. the timestamps of milestones during the port call) and run the consensus mechanism that decide on the true state of the system. The owners of these nodes naturally include the two parties to the charterparty (i.e. the disponent shipowner and charterer/shipper) but also other stakeholders that have an interest in maintaining a copy of the data underlying the smart contract execution. The latter group might include the captain/vessel, cargo terminals, port state control (PSC) authorities and customs agencies. However, depending on the consensus mechanism and network structure (public or private), the nodes may simply also represent third-party cloud-based computing power or block miners. Each node is connected to the platform by authenticating with its own

[3] We acknowledge that a broader implementation of smart contracts in the chartering process may make the supply chain entirely autonomous such that the traditional search-offer-acceptance negotiation process no longer exists but is replaced by an algorithm, which could be a central platform or decentralized application. We here consider only the narrow implementation of smart contracts in relation to demurrage.

[4] The most well-known examples are perhaps the trading of cryptocurrencies such as Bitcoin and digital assets such as "Cryptokitties"

private key, which also determines the node's authorization level. Figure 1 shows the architecture of the smart contract ecosystem conceptually. We have here differentiated between the storage and verification of primary data on the blockchain, and the execution of the smart contract in a layer built on top. This may be necessitated by the ability of certain blockchain solutions to scale.

We note that while the objective for smart contract implementation would be automated monitoring and execution, we cannot rule out that one outcome would be a dispute (i.e. the nodes cannot reach consensus on the true state of the situation). In this case, the smart contract terminates and the resolution of the dispute is handed back to a pre-selected arbitrator, tribunal, or court in the physical world, as illustrated in Figure 1.

Although automatic settlement of any agreed demurrage payment could be an integral part of the smart contract itself, this is not a key element in our mind. Firstly, payments are not recognized as a pain point in an industry which is relies on large international bank transfers on a daily basis. Secondly, the number and frequency of transactions would be too small to justify a separate digital currency. Thirdly, in order to ensure automated payment, funds would need to be tied up when the parties enter into the contract. Such liquidity requirements – to cater for an event which may not occur – would unnecessarily increase the cost of doing business. The output of our smart contract is therefore simply an agreed calculation of the amount payable.

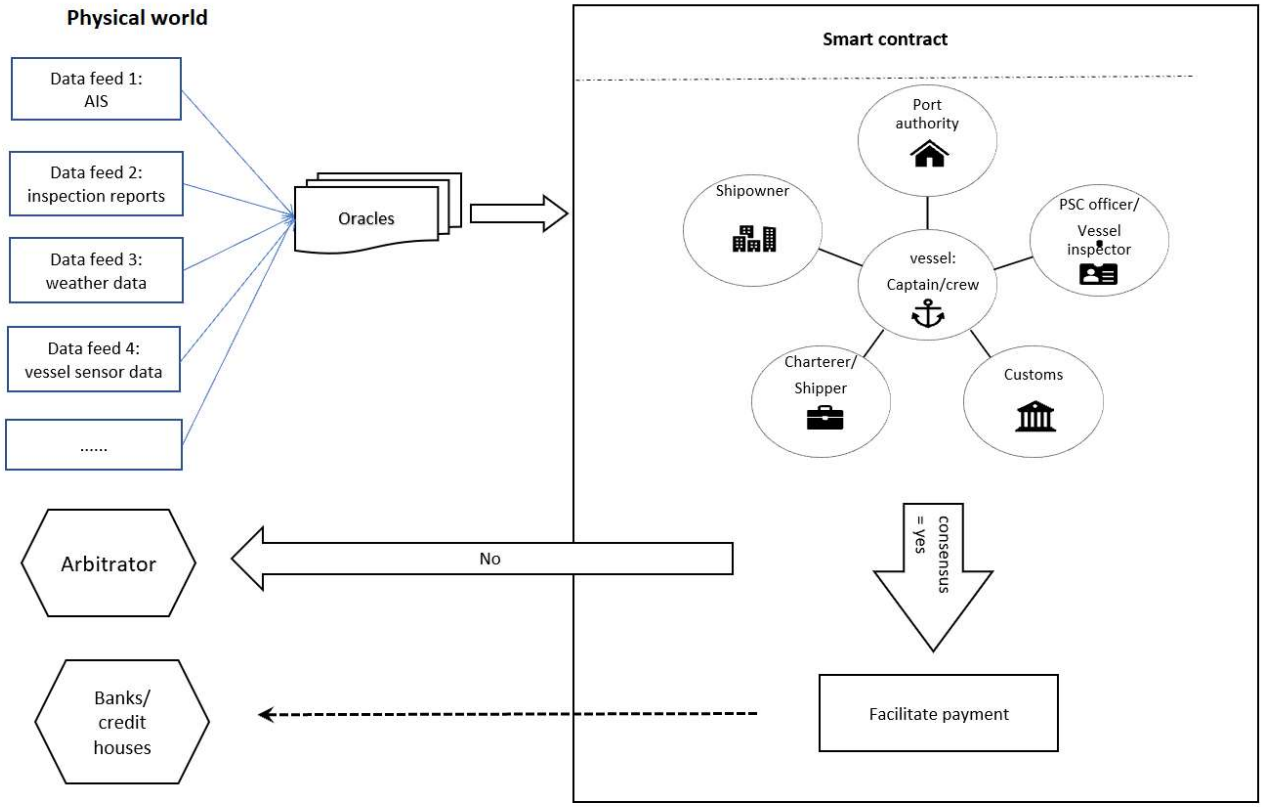


Figure 1. Architecture of the smart contract ecosystem

3.3 Verification and consensus mechanisms

With this ecosystem in mind, let us consider how the conditions for demurrage to occur, as outlined in Chapter 2, could be monitored and verified by our smart contract. The objective would ultimately be to break down and standardize the legal language into conditions that can objectively be categorized by a computer algorithm. Let us consider the relevant charterparty clauses in turn.

3.3.1 Whether a ship has reached the destination

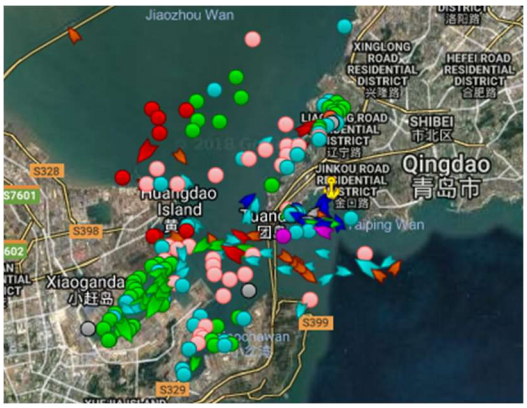
In principle, the existence of ship position data from the global Automated Identification System (AIS) enables remote continuous tracking of all ocean-going vessels over time. In this manner, port waiting areas and terminals can be mapped electronically – usually by way of algorithms clustering observations of stationary vessels inside polygons. The timestamp of a ship entering the polygon therefore defines the time of arrival at the destination. Which

geographical areas that constitute the destination can be pre-selected in the contract, and can easily account for the wide definition in BIMCO (2013), "any area where vessels load or discharge cargo and shall include, but not be limited to, berths, wharves, anchorages, buoys and offshore facilities as well as places outside the legal, fiscal or administrative area where vessels are ordered to wait for their turn no matter the distance from that area." Indeed, from observing the past operational behavior of similar ships in the port, such vague language can be made objective and executable by computer code.

3.3.2 Whether a ship is 'reachable'

The commonly used 'Reid test' for whether a ship is reachable and at the immediate and effective disposition of the charterer can be summarized as: if "she is at a place where waiting ships usually lie, she will be in such a position" (c.f. footnote 1). Accordingly, the empirical observation, based on historical AIS data, of where similar ships are waiting will generally be sufficient evidence to determine whether this clause is satisfied.

Figure 2 shows a snapshot of ship traffic in the Chinese port of Qingdao, courtesy of Marinetraffic.com. The port waiting areas and cargo handling terminals for different types of ships are clearly visible.



Notes: Snapshot of vessel traffic in the port from Marinetraffic.com. Circles indicate stationary vessels and arrows indicate moving vessels.

Figure 2. AIS reported ship traffic in the port of Qingdao

3.3.3 Whether a ship is 'ready'

Clearly, the assessment of whether a ship is ready to load or discharge is, for now, beyond the possibilities of algorithms based on onboard sensor data, electronic logs or AIS data. Assessing readiness thus becomes a matter of trust in the opinion or information given by an "off chain" resource such as the captain, crew, supercargo or port agents. This is in some sense the antithesis of blockchain as a champion of 'trustless' interaction. However, the blockchain data structure can still facilitate secure sharing of the relevant information. For instance, the AIS data provider can confirm that the vessel has reached the destination port and is waiting at an assigned anchorage, the chief engineer can confirm that the vessel has adequate fuel supplies, and the chief officer and port agent can confirm that the cargo holds and gear are ready for loading. Consensus can then be reached that the ship is ready and the NOR can be tendered electronically on the blockchain, triggering the commencement of laytime.

Another point of contention would be interruption of loading or discharge due to events beyond the control of the charterer, usually due to inclement weather leading to closure of the port or endangering the quality of the cargo. While the occurrence of rain showers, for instance, can be monitored by onboard sensor data, whether the severity of the weather allows for laytime to stop counting remains somewhat subjective.

4. CHALLENGES AND IMPLICATIONS

As noted by Levi and Lipton (2018), it is quite likely that a court today would recognize the validity of computer algorithms that execute provisions of a traditional contract, such as the demurrage clause within a charterparty, given the existing legal framework for recognizing electronic contracts. The challenge to large scale adoption may, therefore, have less to do with the limits of the law than with the differences between how smart contract code operates and how parties transact business. Levi and Lipton (2018) also point out that blockchain-based smart contracts are not truly "trustless" as a great deal of trust is placed in the programmer translating legal principles and clauses into computer code, not least because recent research (e.g. Nikolic et al., 2018) suggests that technologists still do not have a full picture of what a security hole in a smart contract looks like.

As an important general point, the mere entry of data on a blockchain and generation of consensus based on such data does not guarantee that the data is accurate and trustworthy. The source of most demurrage disputes is, after all, a lack of agreement on the timeline and sequence of events. This could be because of a lack of recorded information altogether, a belief that the recorded information has been tampered with, or a lack of trust in the data quality or the provider of the information. A blockchain-based smart contract can only help with the first two of these trust-related issues: the recording of key data and timestamps, with immutability of the records as a key feature. However, as long as there is an electronic/human/physical interface, data can be entered incorrectly, either due to sensor malfunctioning, typos, miscommunication or human mischief. For instance, it is well known that ship positions and even ship identities reported by the AIS system can be spoofed and manipulated (Katsilieris et al, 2013). A similar problem may arise because of the latency inherent in the satellite communication required for ship-to-shore communication, creating inconsistencies in the timestamps of events. By itself, blockchain, or more generally

distributed ledger technology, therefore cannot completely establish trust in the input data underlying the monitoring of contractual performance and laytime calculations.

This brings us to another key issue – the economic interests of the contractual parties in the chain. The interests of the nodes in the blockchain (c.f. Figure 1) are typically not aligned, creating incentives to misreport, delay data reporting, stall the consensus mechanism or even collude to approve an outcome which is factually wrong. In this manner, merely creating a blockchain for storing and exchanging data does not solve all of the current problems leading to disputes. However, we believe it will facilitate quicker dispute resolution, as a substantial part of the timestamps and data would be hard to argue with. It remains an open question whether a blockchain-based system can be built to better align the interests of the contracting parties, for instance, through the adoption of a token platform that rewards correct reporting and penalizes misbehavior and we leave this for future research.

On a higher level, the entire concept of demurrage is increasingly being questioned, with suggestions being made to abandon the principle altogether. Importantly, from the point of view of "greening" the supply chain, demurrage acts as a contractual barrier to increasing environmental efficiency. For instance, Jia and Adland (2018) show that demurrage has the perverse effect of increasing the optimal sailing speed (and corresponding ship-to-air emissions) in poor freight markets, when the daily profit from claiming demurrage exceeds that of sailing the vessel. In practice, together with the First-in-First-out berth allocation policy in most ports, the demurrage clause encourages the oft-observed "hurry-up-and-wait" behavior in ocean transportation (Psarros, 2017). Trying to increase the efficiency of the demurrage process is therefore akin to treating the symptom rather than the cause.

For managers, the implication is that care should be taken before heavy investment is made in autonomous smart contract applications related to the demurrage process. The challenges related to data quality and incentive systems need to be solved first. However, our analysis also points to the value of digitalizing and storing a common event log, accessible to all stakeholders and based on input from sensor and tracking technology that is already available, as a basis for resolving potential demurrage disputes. However, such a shared electronic log need not be based on blockchain technology. Indeed, given the potential for errors, the immutability of data records could be more of a drawback than a selling point.

5. CONCLUDING REMARKS

The major obstacle facing blockchain-based smart contracts in physical industries such as ocean transportation is the fact that their execution relies heavily on "off chain" resources for the input and verification of data, be it physical sensor data or human input. There simply is no algorithm that can verify the physical status or location of a vessel by mathematical computations alone. The challenging interaction with the real physical world is not unique to our application to smart contracts for demurrage in ocean shipping.

Such reliance on off-chain resources creates a new set of practical challenges in the implementation of smart contracts. Firstly, high error rates in the input data may create incorrect consensus and contract execution, or necessitate multiple data revisions on the blockchain to correct to the 'true state'. Secondly, misaligned interests between the nodes in the distributed system (i.e. contracting parties, their agents, and other off-chain resources), creates incentives to collaborate to trick the system for financial gain (i.e. wrongfully reducing or increasing demurrage payments in this case). Merely relying on the voting of the majority in a blockchain-based consensus mechanism therefore does not guarantee the correct outcome. Thirdly, even it was in everybody's interest to act fairly, the high cost and intrinsic latency of global satellite communication connecting ships, distributed sensors and shorebased agents would create computational difficulties in agreeing on even a simple timestamp.

Many of these challenges can be resolved over time, either through new and improved systems and data protocols for tracking vessel positions, cheaper satellite communication, or the creation of incentive systems (possibly digital tokens) that can align the interests of nodes in the chain or expand the pool of 'oracles' verifying the true status of a vessel.

In the meantime, the main benefit of implementing blockchain technology for demurrage calculation is simply the digitalization and structuring of the data input required, without the autonomous execution that is promised by smart contracts. This may in itself be a benefit that will reduce disputes and increase the efficiency of the supply chain.

With regards to future research, a key part in a successful future implementation of smart contracts for demurrage claims in ocean transportation is clearly the incentive system that needs to be in place to increase accuracy and reduce the potential problems caused by misaligned incentives. Such an incentive system would have to be based on the idea that those who benefit from increased efficiency and reduced demurrage claims (i.e. mainly shippers) would share some of these financial benefits with agents who provide accurate information on the location and status of a vessel and cargo handling operations, likely through digital tokens built on the smart demurrage contracts The economics of such digital tokens is a research area in its infancy, but clearly the logistics industry – with its many pain points related to demurrage costs and delays – is ripe for such innovation.

6. ACKNOWLEDGEMENT

This research was funded by the Research Council of Norway as part of the project 'Smart digital contracts and commercial management', project no. 280684.

7. REFERENCES

[1] AET v. Eagle Petroleum, (2010) 2 Lloyd's Rep. 257

[2] Armement Adolf Deppe v. John Robinson & Co., (1917). 2 K.B. 204

[3] Bag, S. 2018. "Economic Analysis of Contract Law, incomplete contract and asymmetric information." Palgrave Macmillan, Springer International Publishing

[4] Bashir, I. (2018). "Mastering Blockchain: Deeper insights into decentralization, cryptography, Bitcoin, and popular Blockchain frameworks." Packt Publication: Birmingham

[5] BIMCO. 2013. "Laytime definitions for charter parties clause". London: BIMCO.

[6] BIMCO, CMI, FONASBA and INTERCARGO. 1993. "Voyage charter party laytime interpretation rules"

[7] Bonney, J. 2018. "Arbiter or blame", Journal of Commerce. February 2018, 28-30

[8] Cooke, J., Kimball, J.D., Young, T., Martowksi, D., Ashcroft, M., Lambert, L, Taylor,A., and Sturley, M. 2014. Voyage Charters 4th ed. Abingon and New York: Routledge.

[9] Dwork, C. Goldberg, A. and Naor, M. (2003), On memory-bound functions for fighting spam. Advances in Cryptology: CRYPTO 2003. Springer. 2729: 426–444.

[10] EU (European Union). 2014. Regulation (EU) no 910/2014 of the European parliament and of the council, of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC. Official Journal of the European Union, L 257/73

[11] Gencon 1976. "General Charter Conditions, BIMCO." (as revised in 1922, 1976 and 1994)

[12] Gergen, M.P. 1992. The use of open terms in contract. Berkeley Law Scholarship Repository, (92) Colum. L. Rev. 997, 1006

[13] Glaser, F. 2017. "Pervasive Decentralization of Digital Infrastructures: A Framework for Blockchain enabled System and Use Case Analysis." In, Proceeding of the 50th Hawaii International Conference on Systems Science (HICSS) http://scholarspace.manoa.hawaii.edu/bitstream/10125/41339/1/paper0190.pdf)

[14] Hadfield, G.K. 1984. "Judicial competence and the interpretation of incomplete contracts." Journal of Legal Studies, 23(1): 159-184

[15] Hansen, J.D., L. Rossini, and C.L. Reyes (2018). More legal aspects of smart contract applications. https://www.virtualcurrencyreport.com/wp-content/uploads/sites/13/2017/05/Perkins-Coie-LLP-Legal-Aspects-of-Smart-Contracts-Applications.pdf

[16] Jia, H. and Adland, R. 2018. "Vessel speeds: explaining the gap between theory and practice." Annual conference of the International Association of Maritime Economists (IAME), September, Mombasa, Kenya

[17] Katsilieris, F., Braca, P. and Coraluppi, S. 2013. "Detection of malicious AIS position spoofing by exploiting radar information." In Information fusion (FUSION), 2013 16th international conference on (pp. 1196-1203). IEEE.

[18] K/S Arnt J. Moerland v. Kuwait Petroleum Corporation of Kuwait (1988)

[19] Laffaye, J.P. 2013. "Laytime language revised." Tradewinds News. www.tradewindsnews.com/drycargo/323408/laytime-language-revised

[20] Levi, S.D. and Lipton, A.B. 2018. "An Introduction to Smart Contracts and Their Potential and Inherent Limitations." https://corpgov.law.harvard.edu/2018/05/26/an-introduction-to-smart-contracts-and-their-potential-and-inherent-limitations/

[21] Mongelluzzo, B. 2000a. "The Cost of Idle Containers." Journal of Commerce: http://www.joc.com/maritime-news/cost-idle-containers_20000806.html

[22] Mongelluzzo, B. 2000b. "LA, Long Beach Act to Keep Containers Moving." http://www.joc.com/maritime-news/la-long-beach-act-keep-containers-moving20000725.html

[23] Mongelluzzo, B. and J. Bonney. 2014. "Demurrage outcry." Journal of Commerce: 15(22): 24

[24] MSC Mediterranean Shipping Company SA v. Cottonex Anstalt (2016) EWCA Civ 789

[25] Nikolic, I., Kolluri, A., Sergey, I., Saxena, P., and Hobor, A. 2018. "Finding the greedy, prodigal, and suicidal contracts at scale." Cornel University Computer Science arXiv:1802.06038 [cs.CR]

[26] Oldendorff (E.L.) & Co. G.m.b.H. v Tradax Export S. A. (1973) 2 Lloyd's Rep. 285 at p. 291 (H.L.) The Johanna Oldendorff.

[27] Psarros, G. A. 2017. "Energy efficiency clauses in charter party agreements, legal and economic perspectives and their application to ocean grain transport." Springer: Switzerland

[28] Risius, M. and Spohrer, K. 2017. "A blockchain research framework: what we (don't) know, where we

go from here, how we will get there." Business Infrastructure System Engineering 59(6): 385-409.

[29] Shavell, S. 2003. "Economic analysis of commercial law." NBER working paper. http://www.nber.org/papers/w9696

[30] Sofial v. Ove Skou Rederi, (1976). 2 Lloyd's Rep. 205

[31] Surrey Shipping v. Compagnie Continentale, (1978). 2 Lloyd's Rep. 154

[32] Summerskill, M. B. 1989. "Laytime." Fourth edition. London: Stevens & Sons Limited.

[33] Szabo, N. 1994. "Smart Contracts." http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html (accessed 07/16/2018).

[34] Szabo, N. 1994. "Smart Contracts: Building Blocks for Digital Markets." http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html (accessed 07/16/2018).

[35] UETA 1999. Uniform Electronic Transactions Act (Unif. Law Comm'n 1999) http://www.uniformlaws.org/shared/docs/electronic%20transactions/ueta_final_99.pdf.

[36] Veenstra, A. 2015. "Ocean transport and the facilitation of trade." In: Lee, C.-Y., Meng, Q. (Eds.), Handbook of Ocean Container Transport Logistics: Making Global Supply Chains Effective. Springer, 429–450.

Author index